



**Universitat Autònoma
de Barcelona**

Estudio de transductores piezoeléctricos para la generación de imágenes.

Trabajo Final de Grado

Autor: Jose Espinosa Atero

Tutora: Arantxa Uranga del Monte

Abstract

<p>Los sistema microelectromecánicos es una tencologia que se presenta como una de las más prometedoras ya que tiene el potencial de revolucionar los dispositivos industriales, médicos y de consumo, especialmente en el caso de los transductores micromecanizados de ultrasonidos. El objetivo de este trabajo es el estudio de PMUTs para la generación de imágenes en multiples aplicaciones. Para conseguirlo se ha llevado a cabo un estudio acerca de las geometrías y configuraciones de los PMUTs donde se estudia la presión del haz acústico así como la resolución para ver si es posible el objetivo marcado. Finalmente, se escoge la geometría que más se adecúe a nuestro fin y se presenta un trabajo futuro que podría mejorar ésta última.</p>

Índice

1	Introducción.	6
1.1	Sistemas microelectromecánicos.	6
1.2	Transductores de ultrasonido.	7
1.3	Transductores de ultrasonido micromecanizados.	8
1.4	Objetivo del proyecto.	10
2	Estudio mediante simulación de un PMUT circular.	12
2.1	Introducción al software de simulación: Field.	12
2.2	Conceptos previos.	12
2.3	Análisis de un transductor circular.	14
2.4	Conclusiones.	22
3	Estudio mediante simulación de arrays lineales de PMUTs.	24
3.1	Array lineal.	24
3.2	Conclusiones.	28
4	Array rectangular.	30
4.1	Array rectangular sin beam-forming.	30
4.1.1	Conclusiones.	32
4.2	Array rectangular aplicando beam-forming por filas.	33
4.2.1	Conclusiones.	38
4.3	Array rectangular aplicando beam-forming a cada PMUT . .	39
4.3.1	Conclusiones.	41
5	Conclusión y trabajo futuro.	43
6	Referencias.	44
7	Anexos.	46
7.0.1	Presión en función de la profundidad en un único PMUT con forma de pistón	46
7.0.2	Presión en función de la frecuencia en un único PMUT con forma de pistón:	49
7.0.3	Presión en función de la diámetro en un único PMUT con forma de pistón y la directividad:	52
7.0.4	Presión en función del eje 'z' en único array PMUT lineal	55

7.0.5	Presión en función de la diámetro y la directividad en un array PMUT lineal:	58
7.0.6	Presión en en función de la profundidad en array rectangular sin beam-forming.	61
7.0.7	Presión en un array rectangular sin beam-forming y diagrama de presión.	63
7.0.8	Presión en en función de la profundidad en array rectangular con beam-forming por filas.	65
7.0.9	Presión array rectangular con beam-forming por filas y diagrama de presión.	69
7.0.10	Presión en en función de la profundidad en array rectangular con beam-forming individual.	73
7.0.11	Presión array rectangular con beam-forming individual y diagrama de presión.	75

Índice de figuras

1	<i>Estructura y capas de un transductor de ultrasonido piezoeléctrico. Imagen de Piezo-electric Micromachined Ultrasound Transducer (PMUT) Arrays for Integrated Sensing, Actuation and Imaging, Sensors 2015, 15, 8020-8041</i>	8
2	<i>Estructura de un transductor de ultrasonido piezoeléctrico 2. Imagen de Piezoelectric Micromachined Ultrasound Transducer (PMUT) Arrays for Integrated Sensing, Actuation and Imaging, Sensors 2015, 15, 8020-8041</i>	9
3	<i>Estructura de un transductor de ultrasonido piezoeléctrico 3. Imagen de Piezoelectric Micromachined Ultrasound Transducer (PMUT) Arrays for Integrated Sensing, Actuation and Imaging, Sensors 2015, 15, 8020-8041</i>	10
4	<i>a) Huella dactilar a detectar. b) Huella dactilar obtenida mediante ultrasonidos. c) Sistema para detectar la huella.</i>	10
5	<i>Diferencias entre Fraunhofer diffraction and Fresnel diffraction. Imagen de Wikipedia, Campo cercano y Campo lejano, 2019.</i>	13
6	<i>Resolución axial y lateral. Imagen de: Resolution and different transducers. Raghy Kishore Galla, 2015.</i>	14
7	<i>Representación del radio 'r' así como los ejes 'x', 'y' y 'z'.</i>	15
8	<i>Presión normalizada en el eje 'z' para un pistón de diámetro=0.05 y 0.2 mm.</i>	16
9	<i>Presión normalizada en el eje 'z' para un pistón de diámetro=1mm.</i>	16
10	<i>Presión normalizada en el eje 'z' para un pistón de diámetro=2mm.</i>	17
11	<i>Presión normalizada en el eje 'z' para un pistón de diámetro=5mm.</i>	17
12	<i>Presión normalizada en dB en el eje 'z' para $D = 1\text{mm}$ y $D = 2\text{mm}$.</i>	18
13	<i>Presión normalizada en el eje 'z' comparando los resultados para diferentes diámetros.</i>	19
14	<i>Haz de presión para diferentes frecuencias del transductor.</i>	20
15	<i>Haz de presión para diferentes diámetros del transductor.</i>	21
16	<i>Directividad del pistón en función del diámetro.</i>	22
17	<i>a)Representación de un array lineal de siete elementos. b)Propagación de la presión en el eje 'z' a la derecha.</i>	25
18	<i>a)Representación de un array lineal de cinco elementos. b)Representación de un array lineal de nueve elementos.</i>	25
19	<i>Propagación de la presión en el eje 'z' para los diferentes arrays implementados.</i>	26
20	<i>Comparación de la propagación del haz de presión para los diferentes arrays implementados.</i>	27
21	<i>Comparación de la directividad para los diferentes arrays implementados.</i>	28
22	<i>a)Representación de un array cuadrado de 30x30 elementos. b)Propagación de la presión en el eje 'z'.</i>	30

23	a)Representación de un array cuadrado de 20x20 elementos. b)Propagación de la presión en el eje 'z' a la derecha.	31
24	a)Representación de un array cuadrado de 72x9 elementos. b)Propagación de la presión en el eje 'z'.	31
25	Haz de presión en función del eje 'x'. a) Configuración de 30x30 elementos. b) Configuración de 20x20 elementos. c) Configuración de 72x9 elementos.	32
26	Beam-formin usando grupos de sub-arrays. Pulse-Echo Ultrasound Imaging Using an AlN Piezoelectric Micromachined Ultrasonic Transducer Array. JMEMS 2503336 (December 2015).	33
27	Geometría de un array de PMUTs con sus características.	34
28	a)Representación de un array de 15x9 elementos. b)Comparación de presión normalizada en el eje 'z' con beam-forming.	34
29	Comparacion de la propagacion del haz de presion para los diferentes pitches.	35
30	Diagrama de radiación del array con beam-forming por filas. Eje 'x' en función del eje 'z'.	36
31	Diagrama de radiación lateral del array con beam-forming por filas. Eje 'y' en función del eje 'z'.	36
32	Esquema para calcular delays del beam-forming por filas.	37
33	a)Representación de un array de 15x9 elementos. b)Comparación de presión normalizada en el eje 'z' con beam-forming individual.	39
34	Comparacion de la propagacion del haz de presion para los diferentes pitches.	40
35	Diagrama de radiación del array con beam-forming a cada PMUT.	40
36	Diagrama de radiación lateral del array con beam-forming a cada PMUT.	41

1. Introducción.

1.1. Sistemas microelectromecánicos.

Los sistemas microelectromecánicos o *micro-electromechanical systems* (a partir de ahora MEMS) es una tecnología definida típicamente como dispositivos de dimensiones micrométricas que pueden ser actuados eléctricamente y/o sensados mecánicamente. La primera referencia del poder de la miniaturización y, una de las más esclarecedoras sobre el concepto de los MEMS, fue proporcionada por el Dr. Feynmann en una conferencia en la reunión anual de la *American Physics Society* en 1959. Seis años después se publicó el primer trabajo de MEMS: *The resonant gate transistor*. Pero no fue hasta tiempo después que la tecnología de MEMS despegó debido a problema técnicos y falta de técnicas adecuadas en el momento. Fue a principio de los años noventa cuando varios productos comerciales de MEMS llegaron a estar disponibles en aplicaciones de mercado masivo: impresoras de inyección de tinta y acelerómetros de bolsa de aire. Hoy en día las aplicaciones de MEMS son realmente populares e incluyen detección: presión y giroscopios, aplicaciones químicas y biológicas así como actuadores y micrófonos.

Gracias a los avances en el campo de los semiconductores, los MEMS son una tecnología que puede aplicarse utilizando una gran diversidad de materiales y técnicas de fabricación en función del dispositivo que se quiera fabricar. El abanico tan amplio que supone la incorporación de los MEMS a la tecnología, supone uno de los avances más importantes del futuro ya que cantidad de dispositivos como microsensores, microactuadores, etcétera pueden ser integrados en un mismo microchip.

Los materiales piezoeléctricos han sido usados a lo largo de los años en la mayoría de los resonadores mecánicos debido al efecto piezoeléctrico que presentan: al ser sometidos a tensiones mecánicas, en su masa se adquiere una polarización eléctrica y aparece una diferencia de potencial y cargas eléctricas en su superficie. Este fenómeno también ocurre de forma inversa, ya que los elementos se deforman bajo la acción de fuerzas internas al ser sometidos a un campo eléctrico. El uso de estos dispositivos se ha extendido debido a la cantidad de ventajas que presentan en comparación con otras áreas de aplicación tecnológica: escala de miniaturización, uso de baja potencia, alto rendimiento, buena naturaleza eléctrica, capacidad de funcionar en rangos de tiempo cortos, uso práctico, bajo mantenimiento y la interconexión con diversos sistemas.

Los materiales piezoeléctricos en resonadores son muy comunes y se encuentra en multitud de dispositivos como: osciladores, filtros, sensores físicos y gravimétricos, micrófonos o transductores de ultrasonido. Si se comparan con los dispositivos electrostáticos, los MEMS piezoeléctricos tienen una serie de características que los hacen más interesantes con respecto a los primeros: por un lado, se pueden lograr dispositivos de resonancia de alta frecuencia y temperatura estable con excelente estabilidad de ambas. Esta tecnología resulta útil en el campo de la radiofrecuencia con filtros de alta frecuencia además de que permite una miniaturización significativa. Por otro lado, los sensores

piezoeléctricos no requieren potencia por sí mismos, lo que resulta interesante para sensores de bajo consumo de energía.

1.2. Transductores de ultrasonido.

Un transductor es un dispositivo el cual convierte la energía de una forma en otra, normalmente convierte una señal de energía eléctrica en otra ya sea acústica, mecánica, térmica, entre otras. En el caso de los transductores de ultrasonido, o *ultrasonic transducers*, transforman una energía eléctrica en una acústica si trabaja como actuador y, viceversa para el caso de que actúe como sensor. Un actuador se define como un transductor por el cual un sistema de control actúa sobre un entorno mientras que el sensor, es un transductor que recibe un estímulo o señal de un sistema físico y produce una señal que representa información sobre él.

Estos sistemas, utilizan transductores que trabajan con ondas del rango ultrasónico, que son aquellas ondas acústicas que trabajan por encima de los 18KHz y que son imperceptibles por el oído humano. Estos transductores convierten la energía eléctrica en sonido y luego de recibir el eco, las ondas de sonido se convierten en energía eléctrica que se puede medir y trabajar con ella. Mediante esta tecnología, también se puede detectar objetos o realizar mediciones de distancia punto a punto transmitiendo y recibiendo ráfagas de ultrasonido. Este es un dato interesante ya que este proyecto se basa en el estudio de estos elementos con la finalidad de detectar objetos mediante esta tecnología.

Existen varios métodos para la generación y medición de las ondas de ultrasonido: mediante el efecto electromagnético, acústico, láser y el piezoeléctrico. Este estudio se centra en éste último el cual se basa en un elemento activo compuesto por un material piezoeléctrico.

Otra característica a tener en cuenta que presentan los transductores de ultrasonido es que tienen la máxima deformación en la primera frecuencia de resonancia del elemento activo, que se controla mediante el grosor de la membrana y sus dimensiones. Otra de sus características es el coeficiente de acoplamiento electromecánico, el factor de calidad, la impedancia acústica, la impedancia eléctrica, el perfil del haz acústico, la relación señal a ruido (*Signal to Noise Ratio*), entre otras que se encuentran gobernadas por las propiedades del material y el tamaño del elemento activo.

Analizando el esquema del transductor de ultrasonido, este se compone de una superficie que normalmente se trata de una placa que protege de daños al elemento activo que se encuentra debajo de ésta. Después, la superficie se adhiere al material que amortigua mecánicamente las vibraciones, una vez realizado el proceso de generación acústico. Por otro lado, se puede implementar una capa ubicada entre el elemento activo y la placa que lo protege para variar la impedancia acústica del transductor.

A pesar de las buenas características que presentan los transductores de ultrasonido, se encuentran limitados en algunos aspectos, como por ejemplo la absorción por el medio de la onda generada

por el transductor. A medida que aumenta la frecuencia, la onda generada se encuentra cada vez más atenuada lo que limita las aplicaciones en altas frecuencias. Por otro lado, la implementación de estos dispositivos resulta costosa y se necesita tiempo para llevarla a cabo debido a la tecnología de fabricación disponible. Es por estas limitaciones que se introducen los transductores de ultrasonido micromecanizados que, se basan en la miniaturización de éstos y ofrece unas ventajas significativas respecto a los primeros.

1.3. Transductores de ultrasonido micromecanizados.

Los Transductores de ultrasonido micromecanizados o *Micromachined Ultrasonic Transducer* (a partir de ahora MUTs) son un concepto relativamente nuevo donde se puede entender la realización de estos dispositivos mediante la tecnología MEMS (*microelectromechanical systems*), donde la capacidad de miniaturización del proceso de micromaquinado de silicio ha hecho posible la fabricación de dispositivos para trabajar a frecuencias ultrasónicas. Estos dispositivos utilizan la vibración de flexión como principio de trabajo.

Los transductores de ultrasonido han sido usados para muchas aplicaciones y son famosos por su uso en el sector industrial o en aplicaciones biomédicas. Estos pueden ser excitados mediante diferentes métodos, pero el más común y en el que se basa este trabajo es el efecto piezoeléctrico. Este efecto, resulta interesante ya que es la base de la estructura de los PMUTs que son los dispositivos que se trabajan en este estudio. Un PMUT es una capa de material piezoeléctrico intercalada entre dos capas finas de electrodos de alta conductividad sobre una cavidad con un sustrato de metal rígido.

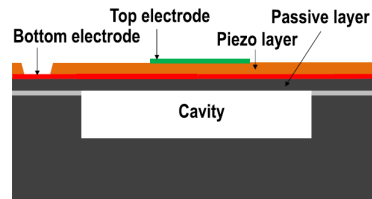


Figura 1: *Estructura y capas de un transductor de ultrasonido piezoeléctrico. Imagen de Piezoelectric Micromachined Ultrasound Transducer (PMUT) Arrays for Integrated Sensing, Actuation and Imaging, Sensors 2015, 15, 8020-8041*

El funcionamiento del PMUT se basa en la transformación entre los dominios eléctrico y mecánico. Más concretamente, cuando se aplica un campo eléctrico a las capas de electrodos, la estructura se dobla provocando la deformación de la membrana debido al efecto piezoeléctrico. Por otro lado, se puede realizar el efecto inverso: a raíz de la deflexión de la estructura y del efecto piezoeléctrico se genera una salida eléctrica. Se diferencian dos tipos de MUTs en función de su principio de actuación:

- *Capacitive Micromachined Ultrasonic Transducers* o CMUTs:

Son los transductores donde la transducción de energía se debe a un cambio en la capacidad, es decir, que el principio de actuación es electrostático.

- *Piezoelectric Micromachined Ultrasonic Transducers* o PMUTs:

Son los transductores donde la transducción de energía se debe a un cambio eléctrico, es decir, que el principio de actuación es piezoeléctrico.

Por otro lado, una característica que presentan los PMUTs es que cada PMUT puede funcionar como transmisor y como receptor. Como transmisor, el campo eléctrico entre los electrodos superior e inferior crea una tensión transversal en la capa piezoeléctrica debido al efecto piezoeléctrico inverso. La tensión generada provoca un momento de flexión que obliga a la membrana a desviarse del plano, lanzando una onda de presión acústica al medio circundante. Como receptor, una presión incidente que desvía la placa crea una tensión transversal que provoca una carga en los electrodos debido al efecto piezoeléctrico directo.

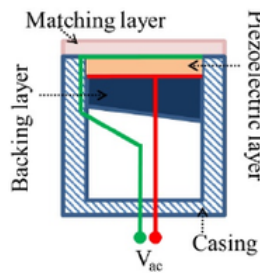


Figura 2: Estructura de un transductor de ultrasonido piezoeléctrico 2. Imagen de *Piezoelectric Micromachined Ultrasound Transducer (PMUT) Arrays for Integrated Sensing, Actuation and Imaging, Sensors 2015, 15, 8020-8041*

Todo sistema físico que se somete a un estímulo puede alcanzar la tasa máxima de absorción de energía, es decir, la frecuencia de resonancia o el grado máximo de oscilación. A la hora de trabajar con PMUTs hay que tener en cuenta la frecuencia de resonancia ya que sino, el sistema implementado podría resultar inestable. Es por eso que la frecuencia de resonancia de los MUTs puede ser controlada mediante una correcta selección del radio del diafragma con una rigidez mecánica efectiva tolerable.

Los transductores de ultrasonido tienen múltiples aplicaciones como por ejemplo ensayos no destructivos de construcciones y naves o en aplicaciones médicas, entre otras. Los sistemas de imágenes por ultrasonido pueden proporcionar imágenes anatómicas y Doppler en tiempo real a alta resolución, y se presentan como soluciones médicas bastante más asequibles que las imágenes por resonancia magnética y la tomografía computarizada. Fuera de las aplicaciones médicas, se han realizados investigaciones abarcando diversas áreas como la ciencia de materiales, acústica, inge-

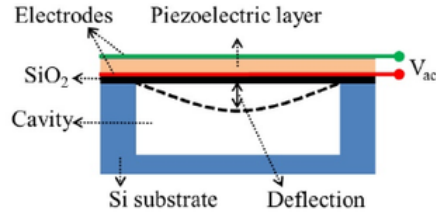


Figura 3: Estructura de un transductor de ultrasonido piezoeléctrico 3. Imagen de Piezoelectric Micromachined Ultrasound Transducer (PMUT) Arrays for Integrated Sensing, Actuation and Imaging, Sensors 2015, 15, 8020-8041

niería eléctrica, ingeniería biomédica, geofísica, predicción de peligros geológicos y alerta temprana del grado de seguridad presas.

1.4. Objetivo del proyecto.

El proyecto que se presenta se ha realizado dentro del grupo ECAS (Electronic Circuits and systems). El grupo ECAS actualmente está desarrollando transductores de ultrasonido implementados sobre circuitería CMOS, utilizando procesos tecnológico de Silterra, que permite obtener una solución monolítica. El PMUT se fabrica directamente sobre la circuitería, haciendo uso de los procesos tecnológicos típicos CMOS. El chip resultante tiene como principales aplicaciones la implementación de un sistema biométrico para la detección de la huella dactilar. El pmut emitirá un pulso de ultrasonido y recibirá el eco de la huella dactilar. En función de la amplitud de la señal recibida el sistema es capaz de reproducir la superficie dactilar (distinguiendo entre las crestas y valles).

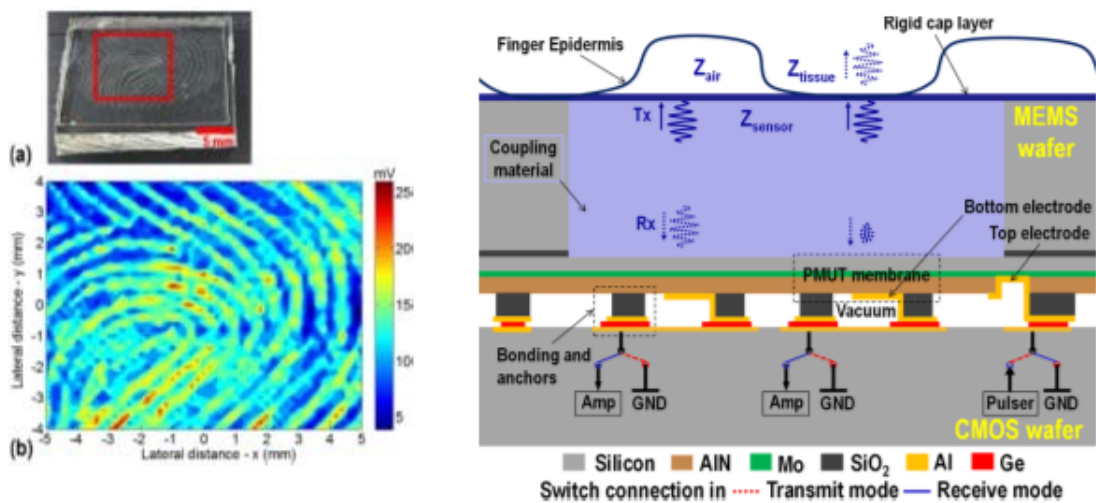


Figura 4: a) Huella dactilar a detectar. b) Huella dactilar obtenida mediante ultrasonidos. c) Sistema para detectar la huella.

La figura anterior es un ejemplo de una huella dactilar obtenida mediante ultrasonidos donde a partir del sistema implementado en c) se obtiene la huella de a) y el resultado se muestra en b). En el sistema se pueden apreciar los PMUTs, una sustancia por la cual viaja el ultrasonido; onda acústica transmitida (Tx) y eco(Rx). El eco, será distinto en función de si se encuentra con un valle o una cresta porque tiene una impedancia acústica distinta, por lo que se obtiene una amplitud más grande y en el otro más pequeña.

El objetivo es realizar un estudio del funcionamiento de los PMUTs como actuadores para poder integrar estos dispositivos en las nuevas tecnologías y avances que se están llevando a cabo hoy en día como es la detección de imágenes, muy empleada en ecografías o en sensores de huellas. Este trabajo se divide en cuatro bloques: en primer lugar, el estudio de un único PMUT circular; en segundo lugar un estudio de un array lineal; a continuación, la implementación de un array rectangular; y, finalmente, se comentan las conclusiones extraídas a partir de los diferentes problemas y resultados obtenidos a lo largo de este proyecto.

2. Estudio mediante simulación de un PMUT circular.

2.1. Introducción al software de simulación: Field.

Field II es un programa para simular campos de transductor de ultrasonido e imágenes de ultrasonido utilizando acústica lineal. El programa utiliza el método Topholme-Stepanishen para calcular los campos de ultrasonido pulsado. El programa es capaz de calcular los campos de eco emitido y de pulso para el caso de onda pulsada y continua para un gran número de transductores diferentes. También se puede simular cualquier tipo de imagen lineal, así como imágenes realistas de tejido humano. El programa se basa en el uso del método *Topholme-Stepanishen*, el cual se basa en el concepto de respuestas de impulso espacial desarrollado por *Topholme* y *Stepanishen*. La respuesta de impulso espacial proporciona el campo de ultrasonido emitido en un punto específico del espacio en función del tiempo, donde el transductor es excitado mediante una función delta-Dirac. La respuesta al impulso variará en función de la posición con respecto al transductor, de ahí el nombre de respuesta espacial al impulso.

La respuesta recibida de una pequeña esfera oscilante se puede encontrar por reciprocidad acústica. La respuesta impulso espacial es igual a la respuesta recibida para una onda esférica emitida por un punto. La respuesta total recibida en el pulso-eco puede, por lo tanto, encontrarse convolucionando la función de excitación del transductor con la respuesta de impulso espacial de la apertura de emisión, con la respuesta de impulso espacial de la apertura de recepción, y luego teniendo en cuenta la transferencia función electromecánica del transductor para producir la traza de voltaje recibida.

2.2. Conceptos previos.

El haz de presión que emana de la placa, no proviene de un único punto sino que se origina en toda la superficie del elemento que se podría dividir en infinitos puntos. Si se analiza la *figura 5*, se puede ver que en cada punto temporal existen algunas regiones esféricas de compresión, que implican el incremento de presión y, alguna regiones esféricas de características más extrañas, que implican la presión reducida. Ambas se encuentran presentes en el plano lo que proporcionará una señal mixta de compresiones y señales extrañas interferentes. La zona donde se encuentran presentes estas interferencias es la zona de campo cercano o de *Fresnel*. Por otro lado, si se observan las ondas en un plano ubicado a algunas *lambdas* de la superficie de la placa, se observa que el plano está completamente contenido en una compresión o descompresión debido a que las regiones de compresión y señales extrañas de interferencia se vuelven casi planas, lo que provoca que no existan interferencias. Esta zona se define como zona de campo lejano o *Franhofer*. En esta región,

el plano se encontrará completamente contenido en un período de media onda de compresión y en descompresión dentro de la otra mitad.

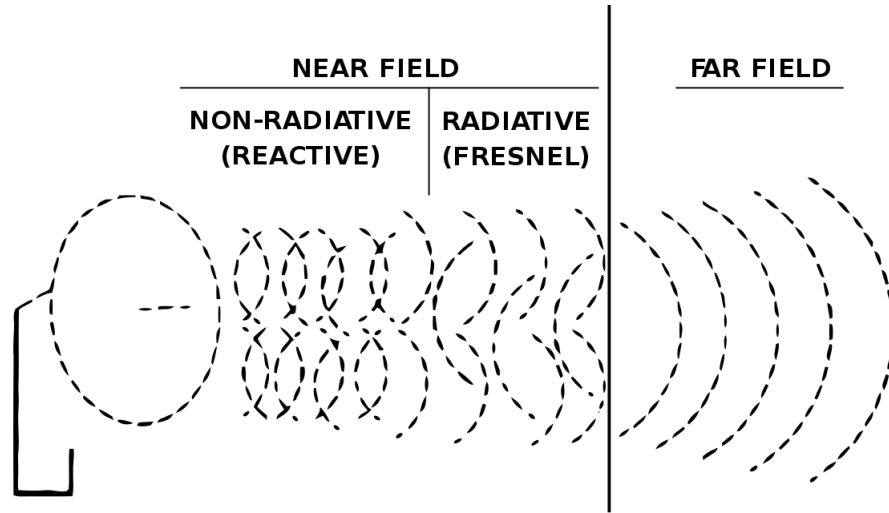


Figura 5: *Diferencias entre Fraunhofer diffraction and Fresnel diffraction. Imagen de Wikipedia, Campo cercano y Campo lejano, 2019.*

Resolución del haz de acústico.

Con el transductor centrado, un haz acústico estrecho es esencial para obtener una alta resolución. La resolución es un parámetro importante que hay que tener en cuenta a la hora de trabajar con PMUTs. Se definen dos tipos de resolución:

- Resolución axial: Es la capacidad de distinguir dos estructuras cercanas entre sí a lo largo del eje central.
- Resolución lateral: Distancia mínima a la que dos estructuras están separadas de lado a lado o perpendiculares al haz de sonido que producen dos ecos distintos.

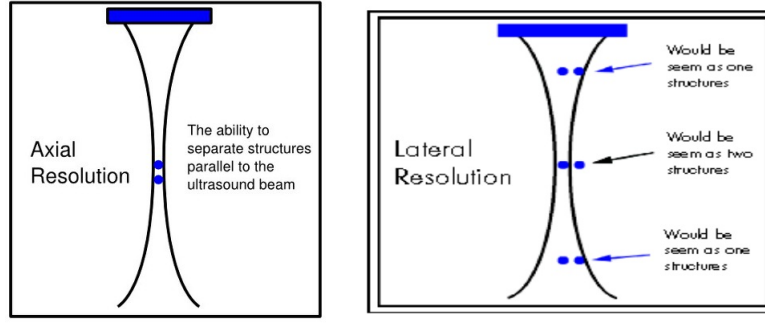


Figura 6: *Resolución axial y lateral. Imagen de: Resolution and different transducers. Raghy Kishore Galla, 2015.*

Una buena resolución es importante para implementar sistemas como por ejemplo, los *finger-prints*. La resolución se define como el número de píxeles por pulgada (o ppp) que caracteriza las imágenes. Intuitivamente, cuanto mayor es la resolución del escáner, más finos son los detalles que se pueden extraer de una imagen de huella digital. Por ejemplo, el sistema compatible con FBI requiere una resolución de 500 ppp. Es por eso que en nuestro estudio, se buscará siempre obtener una buena resolución.

2.3. Análisis de un transductor circular.

En este primer capítulo, se estudia el problema de radiación de un pistón circular excitado uniformemente. Para empezar, se estudia la presión que ejerce el pistón circular en función de la profundidad para poder analizar el campo cercano y el campo lejano. Definir la frontera en la que se puede considerar que nos encontramos lo suficientemente alejados de nuestro dispositivo como para decir que estamos en la región de campo lejano, no es tan directo. Por decirlo de una manera simple, esta distancia dependerá básicamente de nuestra implementación. Es por eso que a continuación, se analizará la distancia de Rayleigh para determinar el campo cercano y el campo lejano y determinar un punto concreto en el que trabajar. En primer lugar, se define un pistón circular con una frecuencia de muestreo de $f_s = 1GHz$, una λ en función de la velocidad del líquido *fluorinert* donde $c = 750m/s$ y una frecuencia centrada del transductor de $f = 8MHz$.

Distancia de Rayleigh.

El modelo general para la presión producida por la vibración del piston montado en un deflector rígido es una solución llamada integral de Rayleigh:

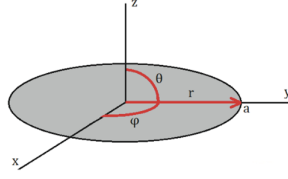


Figura 7: Representación del radio 'r' así como los ejes 'x', 'y' y 'z'.

$$p(x, y, z; t) = \rho_0 \int_S \frac{u_p(x', y'; t - R/C_0)}{2\pi R} dS \quad (1)$$

Donde x' y y' son las coordenadas del punto donde el piston empieza a radiar y R , la distancia hasta el punto de interés que se define como:

$$R = \sqrt{(x - x')^2 + (y - y')^2 + z^2} \quad (2)$$

Pero la ecuación que deriva de (1) y que resuelve los problemas de radiación de un pistón viene dada:

$$p(x, y, z; t) = \frac{jK\rho_0 c_0 \mu_0 e^{j\omega t}}{2\pi} \int_S \frac{e^{jkR}}{R} dS \quad (3)$$

Donde mediante una serie de cálculos se obtiene:

$$p = \frac{jka\omega\rho_0 c_0 u_0}{r} J_0(ka \sin(\theta)) e^{j(\omega t - kr)} \quad (4)$$

$$D(\theta) = \frac{P(r, \theta)}{P(r, 0)} \quad (5)$$

Finalmente, (5) es la ecuación que describe la directividad característica de la fuente. Esta ecuación se define así ya que, por norma general, el pico de radiación ocurre cuando $\theta = 0$. En caso de que el pico de radiación ocurriese en otro ángulo (θ_{max}), el factor directividad se define como $\frac{p(\theta)}{p(\theta_{max})}$. Finalmente, para evaluar la directividad para un pistón, el resultado es:

$$D(\theta) = J_0(ka \sin(\theta)) \quad (6)$$

Otro asunto que se presentan en la radiación de pistones es la distancia de Rayleigh, la cual nos define el punto frontera entre el campo cercano y el campo lejano. Ésta se define como:

$$R_0 = \frac{S}{\lambda} \quad (7)$$

Donde el parámetro S es el área del piston y λ la longitud de onda. En este caso, como se trata de un pistón circular, $\frac{S}{\lambda} = \frac{\pi a^2}{\lambda}$. Así que, finalmente, la distancia de Rayleigh es:

$$R_0 = \frac{ka^2}{2} \quad (8)$$

Donde a es el radio del pistón que se quiere evaluar.

Para analizar la distancia de Rayleigh, se mide la presión que ejerce el pistón descrito en función de diferentes diámetros:

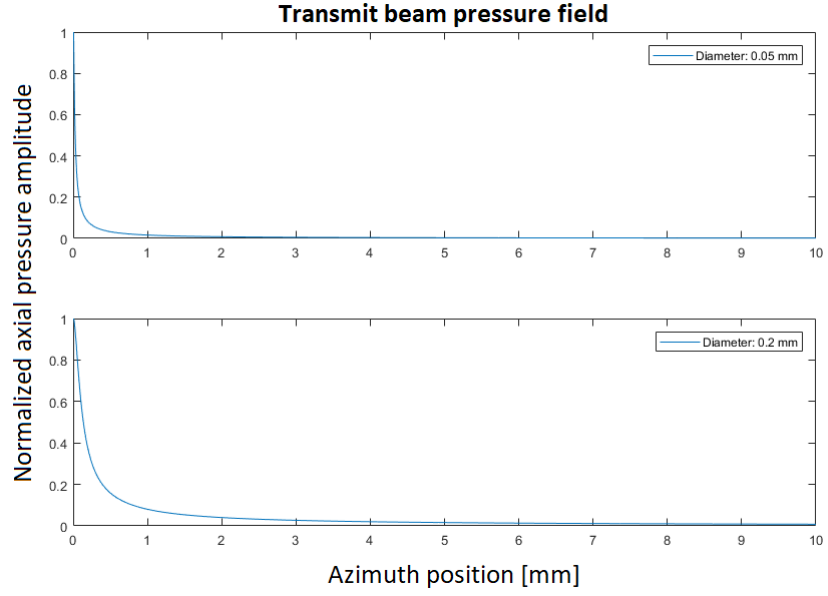


Figura 8: Presión normalizada en el eje 'z' para un pistón de diámetro=0.05 y 0.2 mm.

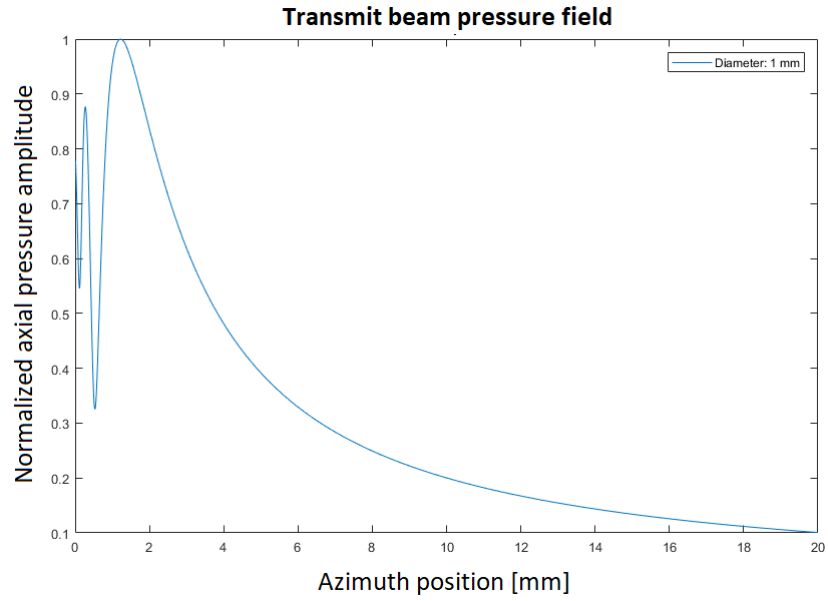


Figura 9: Presión normalizada en el eje 'z' para un pistón de diámetro=1mm.

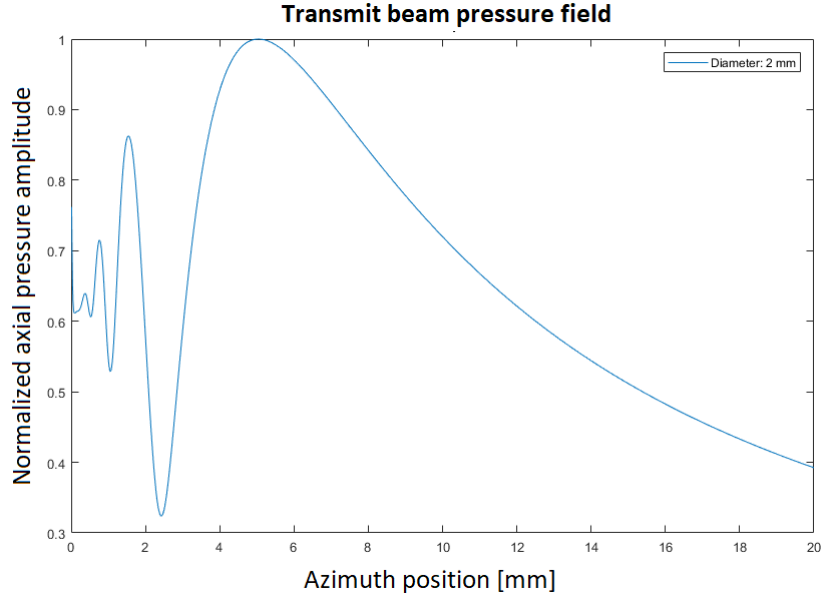


Figura 10: Presión normalizada en el eje 'z' para un pistón de diámetro=2mm.

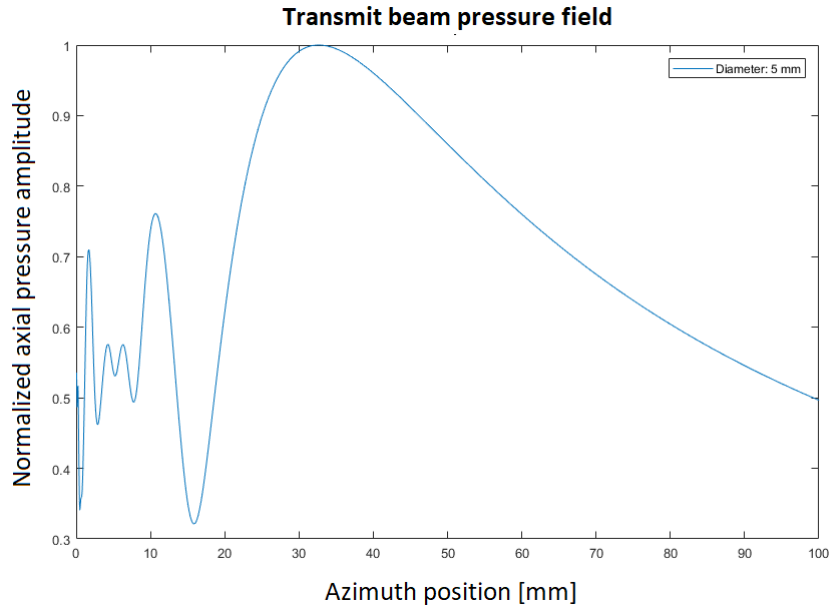


Figura 11: Presión normalizada en el eje 'z' para un pistón de diámetro=5mm.

En las figuras anteriores, se puede ver que después de unas pequeñas oscilaciones estrechas, aparece un pico más alto y más ancho que rompe con la serie de primeras oscilaciones. Este pico (u oscilación mayor) es el punto aproximado en el cuál se determinaría el inicio del campo lejano debido a que es donde se empieza a tener una dependencia de $1/x$. Es decir, las primeras oscilaciones irregulares se interpretan como el campo cercano, mientras que a partir del punto mencionado, empezaría el lejano.

Analizando el campo cercano, se puede ver como la amplitud varía a lo largo de la profundidad hasta el punto de campo lejano. Si analizamos la presión a lo largo del eje 'z' para el diámetro de $D = 1mm$ y $D = 2mm$ en dBs, ver *figura 11*, se puede apreciar que la amplitud en el campo cercano presenta una serie de picos y nulos. Estos picos y ceros en el campo cercano vienen determinados por el valor del radio entre la longitud de onda :

$$\frac{a}{\lambda} \quad (9)$$

Si el resultado de (9) es un numero entero, éste es el número de nulos que presenta la configuración, presentando el primero en el propio pistón. Si por el contrario no es un entero, el primero nulo es el número entero inferior el resultado y el primer nulo se presenta justo en frente del pistón. Para que existan nulos en la eje axial, el radio del pistón debe exceder la longitud de onda. Finalmente, analizando la figura siguiente se puede ver el efecto de doblar el radio del pistón:

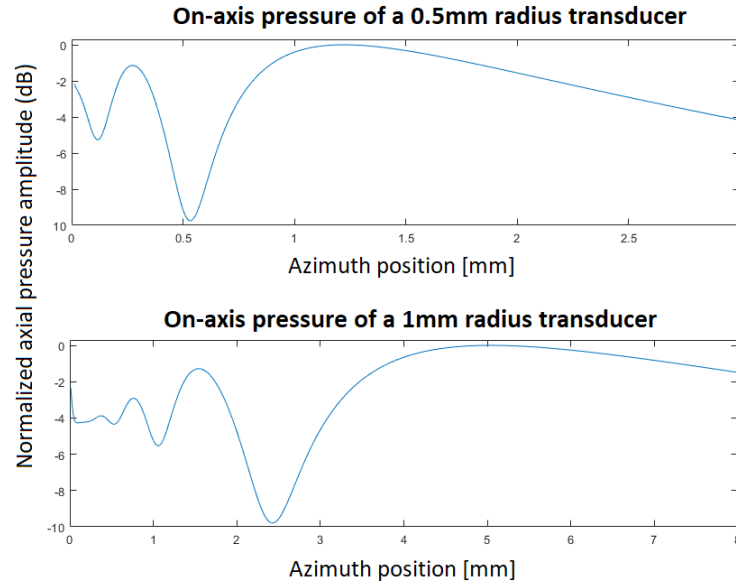


Figura 12: Presión normalizada en dB en el eje 'z' para $D = 1mm$ y $D = 2mm$.

Para corroborar esto, se realizará una tabla donde se calcula la distancia de Rayleigh para cada caso mostrado en las figuras anteriores. La distancia de Rayleigh se calculará a partir de (8) donde λ es igual:

$$\lambda = \frac{c_{fluorinert}}{f_s} = \frac{750}{10^9} = 7,5 \cdot 10^{-7} \quad (10)$$

Distancia de Rayleigh	
$Diametro(mm)$	$R_0(mm)$
0.05	0.0026
0.2	0.042
1	1.05
2	4.19
5	26.18

Tabla 1: Distancia de Rayleigh calculada para cada caso.

Finalmente, se puede concretar que los parámetros calculados en la tabla anterior coinciden con lo que se muestra en el resultado de los experimentos. Otra conclusión de gran importancia que se puede extraer de los resultados es que, a medida que se aumenta el tamaño del pistón, es decir, cuanto más grande es el diámetro, más lejos se encuentra el campo lejano. Esto se puede ver de una forma más clara en la *figura 13* donde se comparan todos los experimentos mostrados anteriormente.

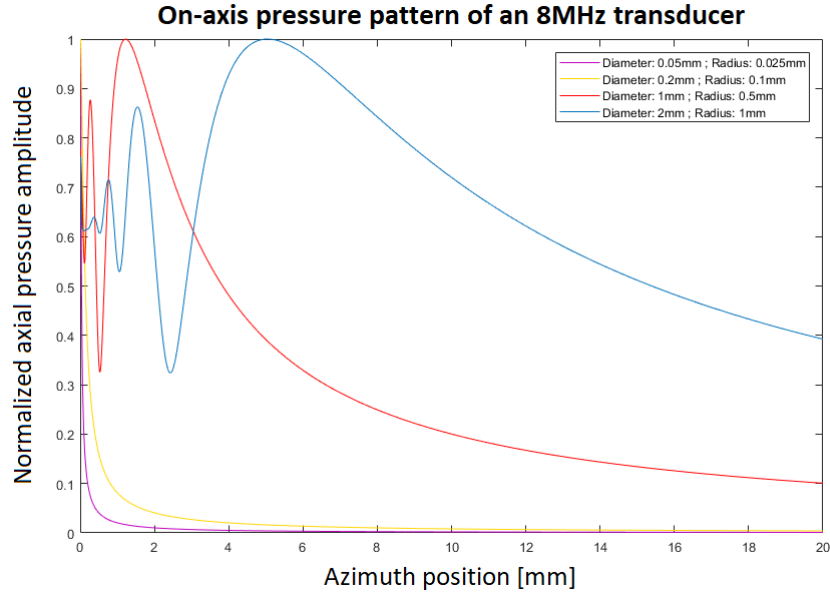


Figura 13: Presión normalizada en el eje 'z' comparando los resultados para diferentes diámetros.

Una vez calculada la distancia de Rayleigh, se realizan experimentos en los que se varía la frecuencia de trabajo del transductor o el diámetro de éste para ver qué efectos producen. Para determinar qué tan buenas son las configuraciones implementadas, se busca que el haz de presión sea lo más estrecho posible y que los lóbulos u oscilaciones laterales sean lo más pequeñas posibles.

En primer lugar, se establece el diámetro del transductor a $200\mu m$, así como la distancia del eje 'z' a $7mm$ (*far-field*) y se realiza el experimento con diferentes frecuencias:

Beamprofile diameter = 200um transductor at depth = 8mm with fs = 1GHz

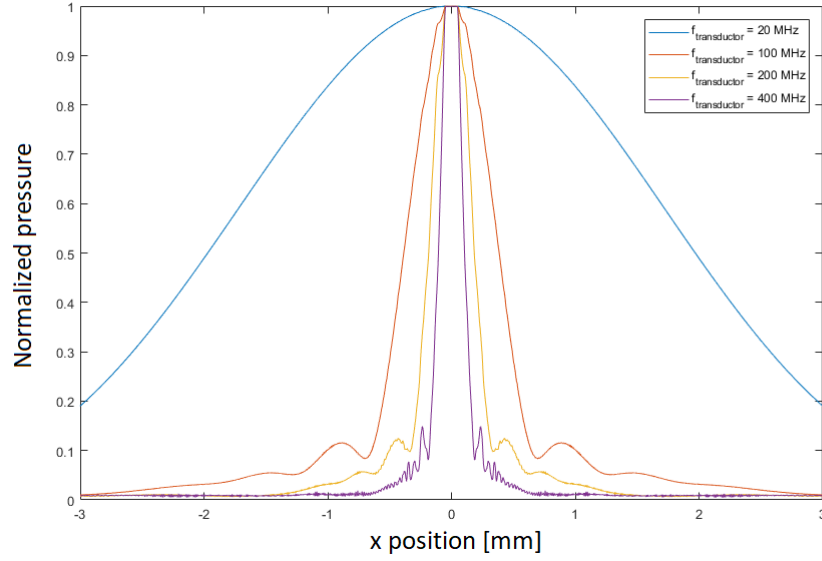


Figura 14: Haz de presión para diferentes frecuencias del transductor.

Ancho del haz acústico a -3dB	
<i>Frecuencia(MHz)</i>	<i>Ancho del haz(mm)</i>
20	3.938
100	0.765
200	0.3676
400	0.21

Tabla 2: Valores ancho del haz a -3dB para diferentes frecuencias del transductor.

Tal y como se puede apreciar en la *figura 14*, cuanto mayor es la frecuencia del transductor, mayor es la selectividad espacial del haz de presión y mas oscilaciones laterales se obtienen. Por otro lado, cuanto menor es la frecuencia, las oscilaciones laterales se reducen e incluso desaparecen. A la hora de implementar y operar con el PMUT, nos interesa trabajar con una frecuencia fija. Es por eso que se realiza el mismo experimento variando los diámetros y manteniendo una única frecuencia para el transductor.

Ahora, para este experimento, manteniendo la distancia del eje 'z' igual que en el caso anterior, se determina una frecuencia centrada del transductor fija de 8MHz y se realiza un barrido de radios:

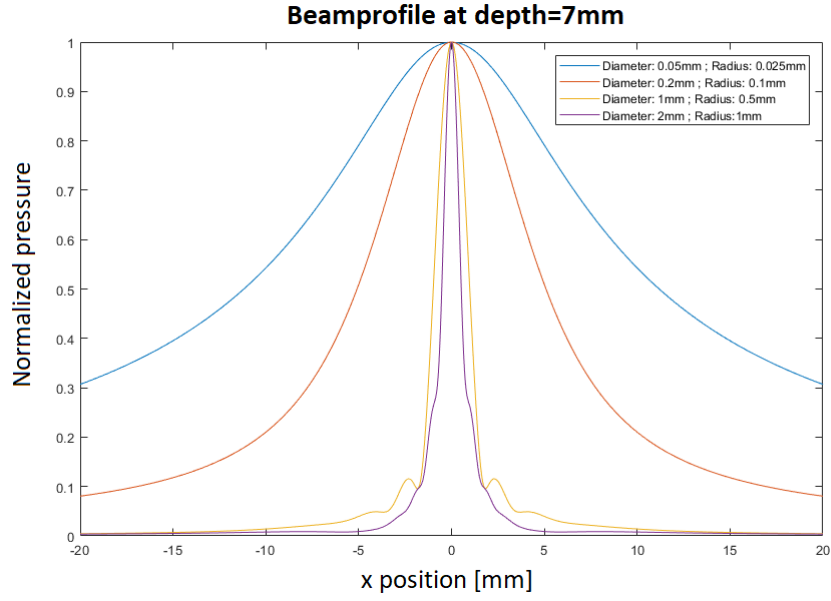


Figura 15: Haz de presión para diferentes diámetros del transductor.

Ancho del haz acústico a -3dB	
<i>Dimetro(mm)</i>	<i>Ancho del haz(mm)</i>
0.05	22.36
0.2	10.15
1	1.95
2	1.05

Tabla 3: Valores ancho del haz a -3dB para diferentes diámetros del transductor.

En la *figura 16*, se puede ver que pasa similar al caso anterior con las frecuencias. Cuanto mayor es el radio del pistón más estrecho es el haz de presión, lo que provoca más directividad. Esto también provoca que aparezcan lóbulos laterales en radios grandes, no así en los que presentan un radio más pequeño. Este resultado se puede analizar más claramente analizando la directividad (antes mencionada). Para ello, se expresa el nivel de presión en función del ángulo de radiación. Partiendo de (4) y aplicando (5) y (6), la ecuación para calcular la directividad de un pistón circular viene:

$$D_{dir}(\theta) = 2J_1 \frac{k a \sin(\theta)}{k a \sin(\theta)} \quad (11)$$

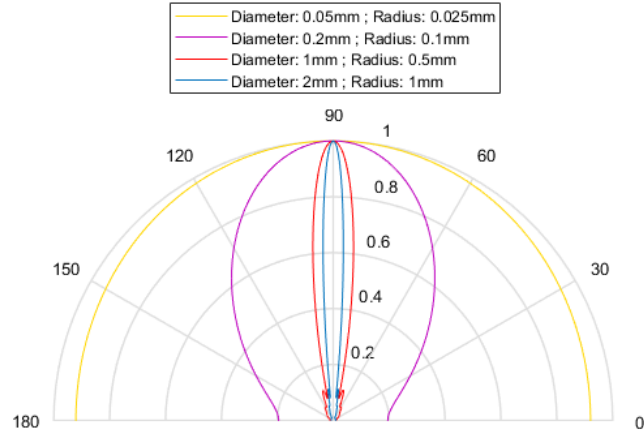


Figura 16: *Directividad del pistón en función del diámetro.*

En la *figura 16* se puede ver la directividad del pistón para cada radio implementado. Esto corrobora, que cuanto mayor es el radio del pistón mayor, directividad tiene.

Directividad	
<i>Dimetro(mm)</i>	$D(\Theta)$
0.05	115.2 ^o
0.2	71.28 ^o
1	15.84 ^o
2	7.92 ^o

Tabla 4: Directividad de un pistón en función del diámetro en grados.

2.4. Conclusiones.

A lo largo de este capítulo se han analizado los efectos de la geometría y frecuencia de un transductor de ultrasonido tipo pistón. En particular se ha analizado:

- Frontera entre campo cercano y campo lejano.
- Diagrama de radiación en función del eje 'x' sobre un punto situado en campo lejano.
- Directividad del PMUT en campo lejano.

Y, las conclusiones de este apartado es que se ha visto como:

- El campo lejano se encuentra a una distancia mayor a medida que se incrementa las dimensiones del pistón.

- El diagrama de radiación en el eje x presenta mayor directividad a medida que se incrementan las dimensiones del piston o a medida que se incrementa la frecuencia de funcionamiento del pistón.
- Como un incremento de la frecuencia del piston va unido a una reducción de las dimensiones del dispositivo. Como solución se opta por el estudio de arrays de pmut.

Por lo tanto hay que estudiar el compromiso entre la relación del haz de presión y la superficie de los PMUTs mediante la implementación de *array PMUTs*.

3. Estudio mediante simulación de arrays lineales de PMUTs.

Debido a las limitaciones encontradas en el caso anterior, en este capítulo se decide trabajar e implementar *arrays* de PMUTs. Los *arrays* de PMUTs son un conjunto de PMUTs ordenados en filas y en columnas los cuales nos permiten determinar el número de elementos que forman dichas filas y columnas para buscar la configuración que más se adapte a nuestros intereses. Todas las simulaciones realizadas en adelante, contarán con una frecuencia de muestreo de $f_s = 1GHz$ y una λ en función de la velocidad del líquido *fluorinert* (donde $c = 750m/s$) y una frecuencia centrada del transductor de $f = 8MHz$. También, contarán con una dimensiones físicas de $50\mu m$ tanto de anchura como de altura y, una distancia entre el centro de los transductores de $73\mu m$. Estos valores han sido escogidos debido a que son con los que trabajan en el grupo ECAS y simulan en *Pulse-Echo Ultrasound Imaging Using an AlN Piezoelectric Micromachined Ultrasonic Transducer Array With Transmit Beam-Forming* Yipeng Lu, Hao-Yen Tang, Stephanie Fung, Bernhard E. Boser, Fellow.

3.1. Array lineal.

En este primer apartado, se realiza una pequeña introducción a los *arrays* PMUTs mediante un *array* de una única dimensión, es decir, un *array* lineal. Se estudian diferentes *arrays* con diferentes configuraciones y finalmente se justifica la selección del más conveniente. Para comenzar, se realiza un *array* de siete elementos y se analizan la propagación de la presión en función de la profundidad, tal y como se muestra en la siguiente figura:

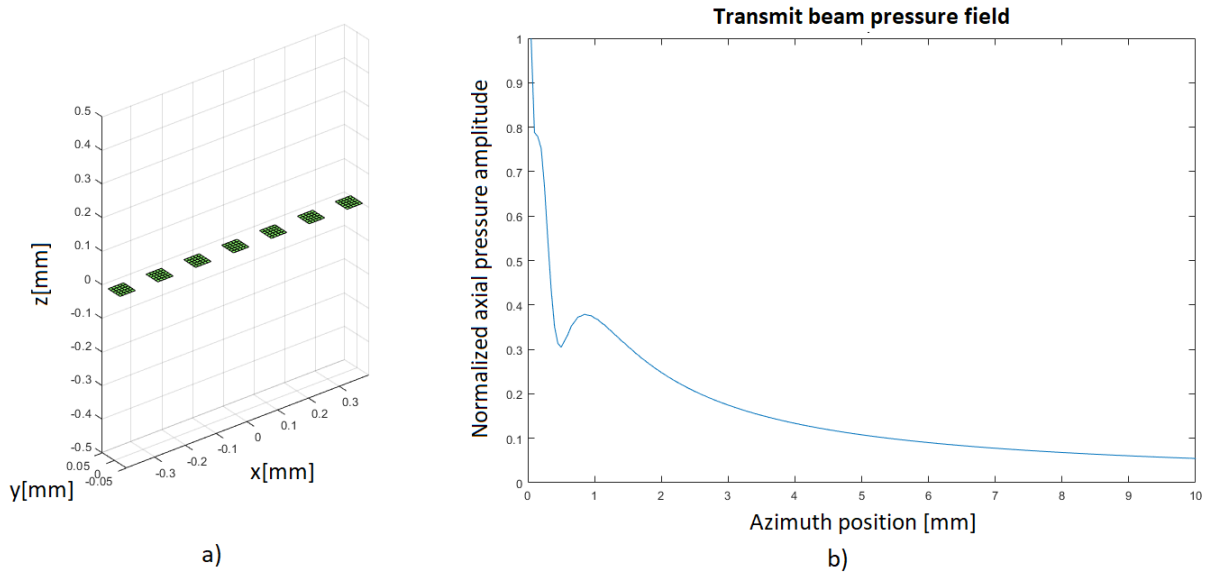


Figura 17: a) Representación de un array lineal de siete elementos. b) Propagación de la presión en el eje 'z' a la derecha.

Como se ha comentado anteriormente, la distancia de Rayleigh se encuentra a partir de la última oscilación grande antes de decrecer, el punto donde se empieza a tener una dependencia de $1/x$. Consideraremos a partir de este punto, el campo lejano. Si se realiza el mismo análisis para dos configuraciones diferentes, una con cinco PMUTs y otra con nueve PMUTs, se observa que el punto de campo lejano se encuentra más cerca en el array con más elementos y más cercano en el de menos elementos.

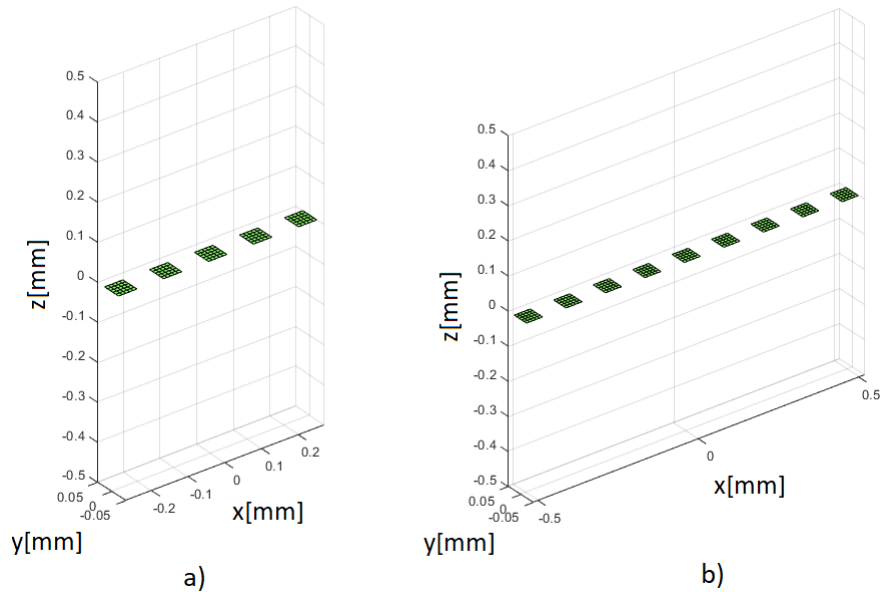


Figura 18: a) Representación de un array lineal de cinco elementos. b) Representación de un array lineal de nueve elementos.

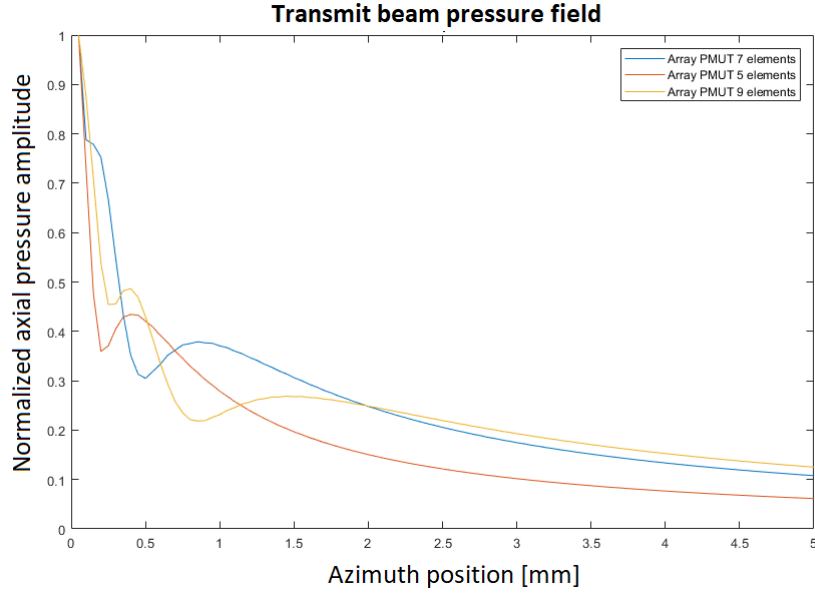


Figura 19: *Propagación de la presión en el eje 'z' para los diferentes arrays implementados.*

Al igual que en el caso anterior, la distancia de *Rayleigh* depende del área global del array dividido entre la longitud de onda. Estos resultados, ya nos pueden anticipar una pequeña información acerca de por qué nos interesa trabajar con arrays de mayor número de dispositivos pero a continuación lo certificaremos realizando un análisis de la propagación del haz de presión y la directividad. Ahora, focalizamos en el punto de campo lejano del *array* más grande y se plotea la propagación del haz de cada configuración focalizada cada una en su respectiva distancia de Rayleigh, tal como se muestra a continuación.

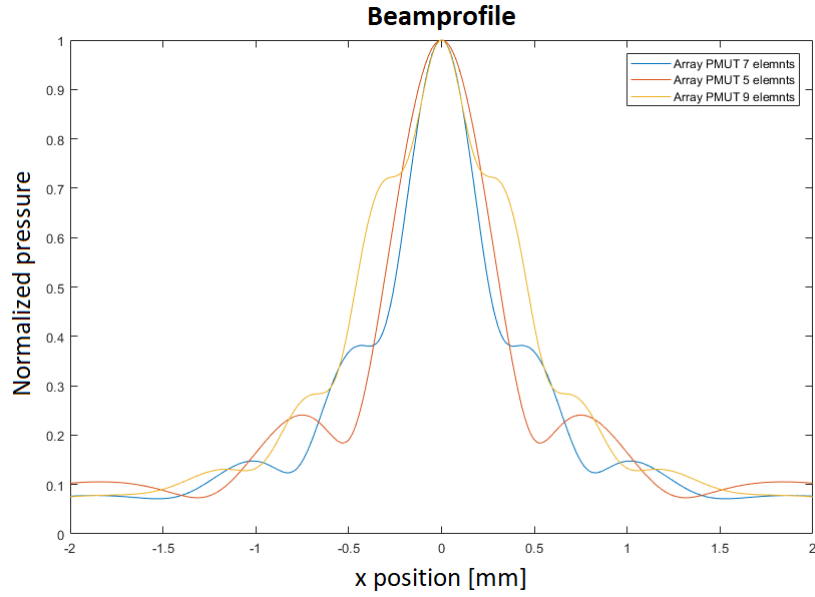


Figura 20: Comparación de la propagación del haz de presión para los diferentes arrays implementados.

Ancho del haz acústico		
<i>N de elementos del array</i>	<i>Ancho del haz(mm)(-3dB)</i>	<i>Ancho del haz(mm)(-0,5dB)</i>
5	0.62	0.24
7	0.52	0.22
9	0.92	0.18

Tabla 5: Valores ancho del hazacústico para diferentes arrays lineales.

Los valores proporcionados en la tabla, se muestra que en el *array* de nueve elementos el ancho del haz a -3dB es más grande que los otros dos pero eso sucede debido a que las oscilaciones laterales aparecen muy tempranamente. Si se analiza el ancho del haz a -0'5dB y antes de que aparezcan las oscilaciones laterales, se puede ver que el ancho del haz es más estrecho cuanto más grande es el número de elementos que se encuentran en el *array*, es decir, cuanto mayor es la superficie ocupada por PMUTs. Tal y como se puede ver en la *figura 20*, se comprueba que cuanto más grande es el número de PMUTs y más superficie ocupen, más estrecho es el haz de presión y, antes aparecen oscilaciones no deseadas. A continuación, se analiza para un mismo caso, la directividad de las configuraciones anteriores.

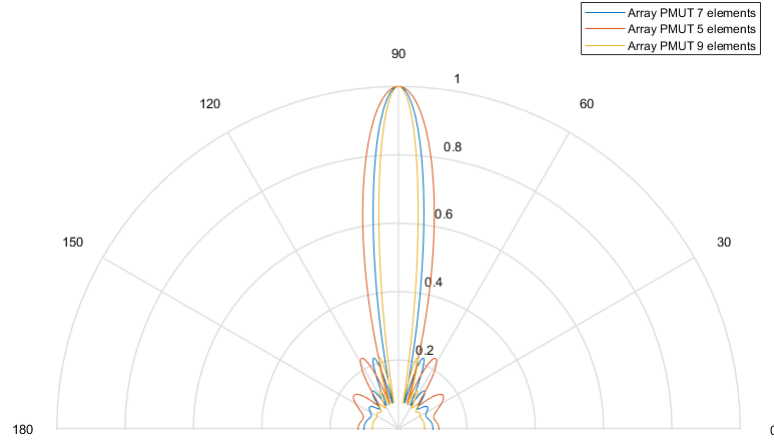


Figura 21: Comparación de la directividad para los diferentes arrays implementados.

En la figura 21, se puede ver la directividad de cada *array*, siendo el de nueve PMUTs el mejor y el de cinco el peor. Esto nos lleva al compromiso de que cuanto más directividad se busque, antes aparecerán los lóbulos laterales que no interesan. De forma que existe un pequeño compromiso entre la directividad y estos pequeños lóbulos. Mediante (5) se puede calcular la directividad de cada configuración de forma matemática a raíz de las simulaciones y corroborar que la configuración que mejor directividad presenta es el de más elementos. Se analiza la presión para el ángulo mínimo ($\Theta = 0$) y la presión en $P = 0,5$ para cada caso, de forma que se obtiene la directividad de cada configuración para un ángulo concreto:

Directividad	
$NdeElementos$	$D(\Theta)$
5	22.33°
7	15.84°
9	12.24°

Tabla 6: Directividad arrays lineales.

Finalmente, se puede corroborar que el haz de presión es menor para la configuración de menos elementos dado un mismo ángulo y, que éste es proporcional al área que ocupen los PMUTs. Es decir que aumenta la directividad si aumenta el área que ocupen los PMUTs y en este caso, aumenta si aumenta el número de PMUTs.

3.2. Conclusiones.

A lo largo de este capítulo se han analizado los efectos de la geometría de un *array* lineal de PMUTs. En particular se ha analizado:

- Frontera entre campo cercano y campo lejano.

- Diagrama de radiación en función del eje 'x' sobre un punto situado en campo lejano.
- Directividad del *array* de PMUTs en campo lejano.

Y, las conclusiones de este apartado es que se ha visto como:

- El campo lejano se encuentra a una distancia mayor a medida que se incrementa el tamaño del *array*.
- El diagrama de radiación en el eje 'x' presenta mayor directividad a medida que se incrementan el número de elementos del *array*.

Debido a la posibilidad de realizar matrices de PMUTs, en el apartado siguiente se estudian los *arrays* de PMUTs donde se verán mejoras significativas respecto a los *arrays* lineales debido a que la superficie con la que se puede trabajar es mayor.

4. Array rectangular.

Para completar el estudio de los PMUTs, en este capítulo se realiza un análisis de matrices de PMUTs en diferentes situaciones. En primer lugar, se realizan una serie de simulaciones con *arrays* excitando todos los elementos al mismo tiempo y se compararán con *arrays* donde se estimulan los PMUTs en tiempos diferentes.

4.1. Array rectangular sin beam-forming.

En este apartado, se realizan diferentes configuraciones de matrices de PMUTs (*arrays* rectangulares) para ver la diferencia que presentan entre ellas y cuál es su comportamiento para nuestro objetivo. Ahora y como se ha hecho en el apartado anterior, se estimulan todos los PMUTs al mismo tiempo.

Primeramente, se realizan tres configuraciones y se analiza la distancia de Rayleigh para cada caso. En primer lugar, se implementa una configuración de 30×30 elementos:

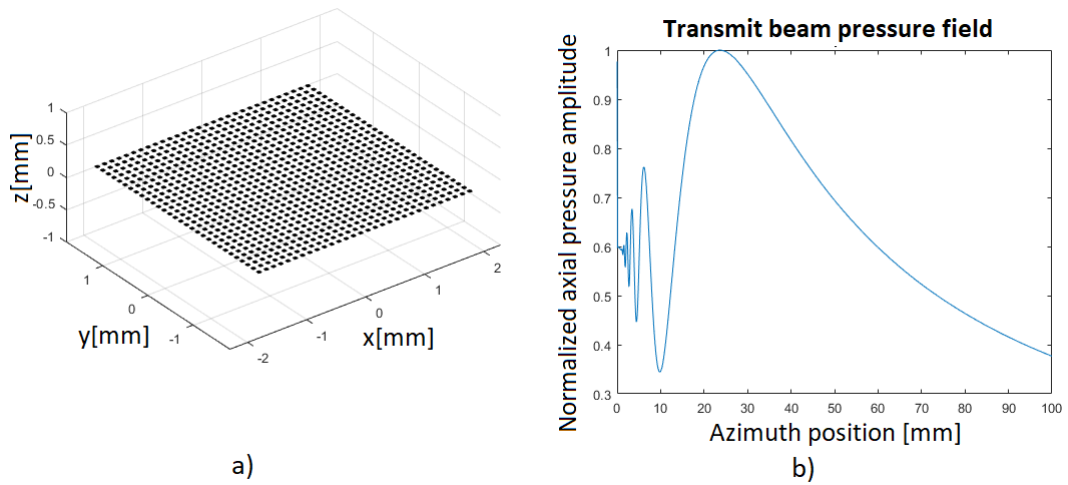


Figura 22: a) Representación de un array cuadrado de 30×30 elementos. b) Propagación de la presión en el eje 'z'.

En segundo lugar, una implementación similar (20×20 elementos) pero con menos elementos para ver las diferencias que presentan el número de elementos cuando el *array* es cuadrado:

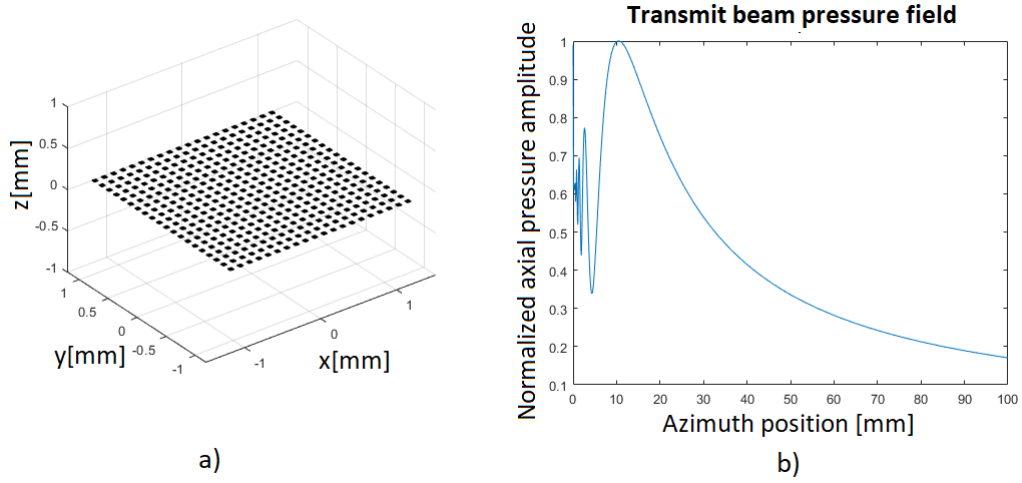


Figura 23: a) Representación de un array cuadrado de 20×20 elementos. b) Propagación de la presión en el eje 'z' a la derecha.

Y, finalmente, un array rectangular (72×9) elementos que se asemeja más a la posibles implementaciones:

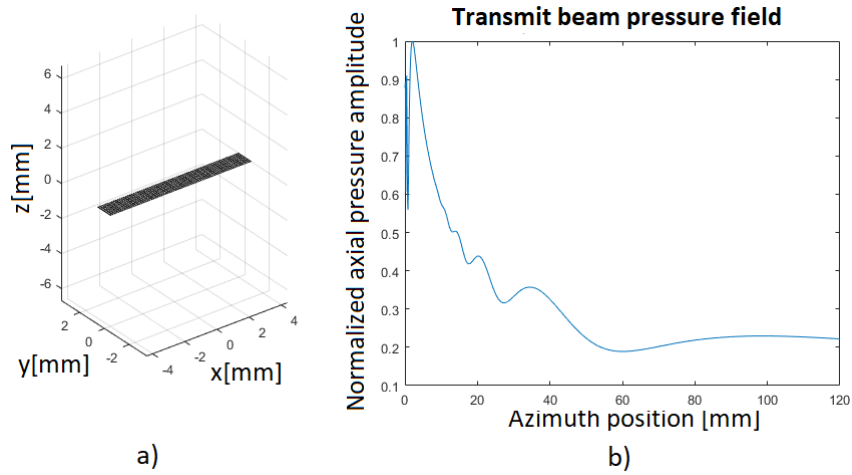


Figura 24: a) Representación de un array cuadrado de 72×9 elementos. b) Propagación de la presión en el eje 'z'.

Comparando los dos primeros, se puede ver lo que se ha visto en los apartados anteriores, y es que cuanto mayor es el área que se cubra con PMUTs, más lejos se encuentra el punto de campo lejano. Si por el contrario, se emplea una configuración con un área más pequeña como éste último caso, la distancia de Rayleigh se encuentra bastante más cercana a cero y los máximos y los mínimos que se presentan en el campo cercano, prácticamente no se pueden diferenciar. A continuación se realizan las simulaciones del haz de presión en función del eje 'x' a una $z = 100\text{mm}$.

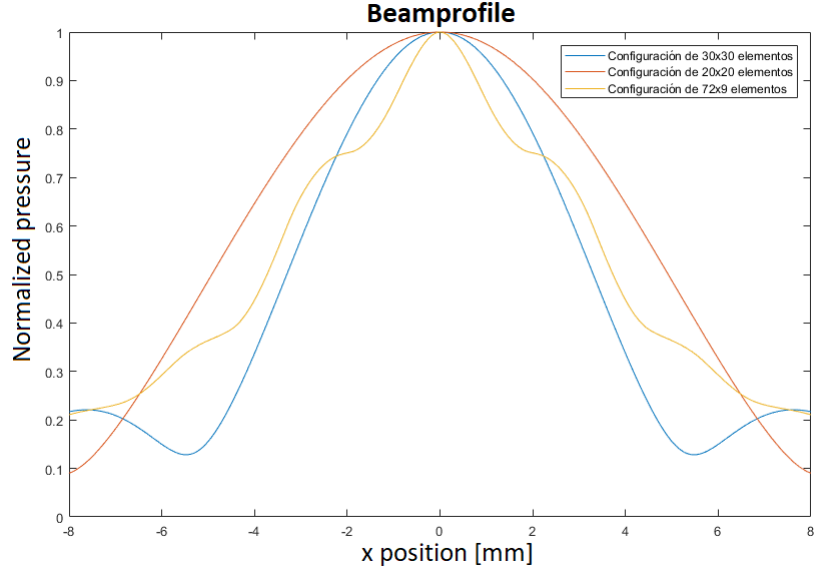


Figura 25: Haz de presión en función del eje 'x'. a) Configuración de 30x30 elementos. b) Configuración de 20x20 elementos. c) Configuración de 72x9 elementos.

Ancho del haz acústico	
Configuración de elementos del array	Ancho del haz(mm)(-3dB)
30x30	6.6
20x20	9.72
72x9	7.42

Tabla 7: Valores ancho del hazacústico para diferentes arrays lineales.

Analizando la *figura 25* donde se excitan todos los PMUTs al mismo tiempo, se puede apreciar que el haz de presión depende de la superficie que ocupen los PMUTs. Para las dos configuraciones de *arrays* cuadrados, tienen un haz de presión bastante similar, siendo éste bastante ancho e impidiendo obtener una buena resolución. Por el contrario, la configuración de 72x9 elementos, presenta una forma diferente debido a los lóbulos lateral que aparecen de forma temprana. Estas representaciones que se presentan impiden obtener una buena resolución debido a la amplitud del haz de presión y que podamos centrarnos en un único punto.

4.1.1. Conclusiones.

A lo largo de este apartado se han analizado los efectos de la geometría y frecuencia de un *array* de PMUTs estimulando todos los elementos a la vez. En particular se ha analizado:

- Frontera entre campo cercano y campo lejano.
- Diagrama de radiación en función del eje 'x' sobre un punto situado en campo lejano.

- Diagrama de presión del *array* de PMUTs cuando se estimulan todos a la vez.

Y, las conclusiones de este apartado es que se ha visto como:

- El campo lejano se encuentra a una distancia mayor a medida que se incrementa las dimensiones del *array*.
- El diagrama de presión que presenta focalización nula cuando se estimulan todos los PMUTs al mismo tiempo.

Para obtener una buena resolución y poder focalizar en un punto, se implementan *arrays* de PMUTs y se aplica *beam-forming* por columnas.

4.2. Array rectangular aplicando beam-forming por filas.

Para solventar los problemas de resolución que existen cuando se estimulan todos los dispositivos al mismo tiempo, se aplica *beam-forming* en los *arrays* de PMUTs. El *beam-forming* es un técnica que se emplea para focalizar el haz de presión en un mismo punto variando temporalmente la excitación de los PMUTs. Consiste en aplicar retardos (*delays* temporales) a los dispositivos de forma que la onda acústica interfiera de forma constructiva en un determinado punto. El *beam-forming* puede ser realizado a nivel de fila o a nivel de PMUT. Si buscamos centrarnos en un punto, es necesario *beam-forming* en cada dispositivo. Si por el contrario buscamos una línea, es necesario aplicar *beam-forming* a nivel de filas. A continuación y para que quede más claro, se muestra en la siguiente figura una representación de *beam-forming* usando grupos de *arrays*.

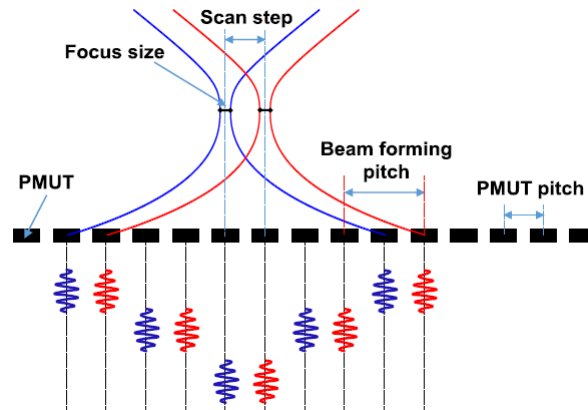


Figura 26: *Beam-forming* usando grupos de sub-arrays. *Pulse-Echo Ultrasound Imaging Using an AlN Piezoelectric Micromachined Ultrasonic Transducer Array. JMEMS 2503336 (December 2015).*

Tal como se muestra en la *figura 26*, si se busca focalizar en el centro del *array* representado, se estimulan los PMUTs que se encuentran más lejanos y, pasado unos segundos, se estimula el elemento siguiente. Así hasta alcanzar el elemento central donde a partir de éste se realiza el proceso

inverso. De esta manera, se consigue un haz centrado en el punto deseado y se puede obtener una buena resolución, solventando los problemas vistos anteriormente.

En nuestro caso, aplicaremos *beam-forming* por filas, es decir, se estudian los tiempos que tiene la columna central de cada fila y se aplica a toda la columna el tiempo registrado por este PMUT. Una vez hecho esto, se realiza el análisis de la propagación del haz de presión en función de la profundidad estimulando con *beam-forming* un *array* de 15×9 elementos en campo cercano. En este apartado, se realizará una comparación de la resolución en función del *pitch* del *beam-forming*, es decir, se varían las columnas que se excitan y se analiza la resolución que presentan.

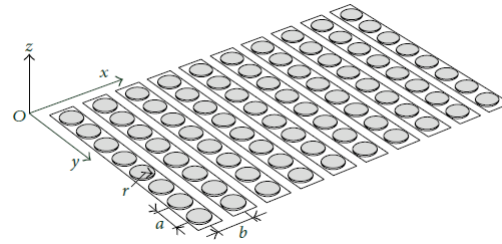


Figura 27: Geometría de un array de PMUTs con sus características.

En primer lugar, se analiza la presión en función del eje 'z' donde se pueden apreciar los máximos en $z = 1,5\text{mm}$ y se focalizará en el centro del *array* en campo cercano y se comparan los efectos que produce aumentar la distancia entre los PMUTs para tres configuraciones distintas: una distancia entre PMUTs de $70\mu\text{m}$, el doble de ésta $140\mu\text{m}$ y por último para $210\mu\text{m}$

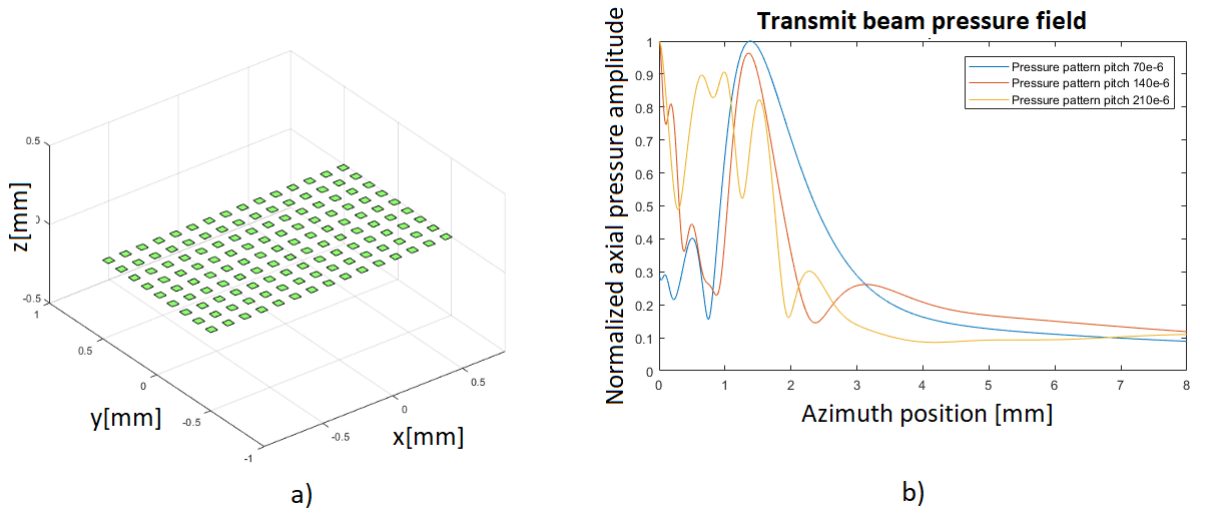


Figura 28: a) Representación de un array de 15×9 elementos. b) Comparación de presión normalizada en el eje 'z' con *beam-forming*.

Al aplicar *beam-forming*, a diferencia de los casos anteriores, la figura 28 nos indica que el

máxima se encuentra en el punto donde hemos aplicado *beam-forming*. Si ahora se analiza la presión en función del eje 'x' para una $z = 1,5mm$.

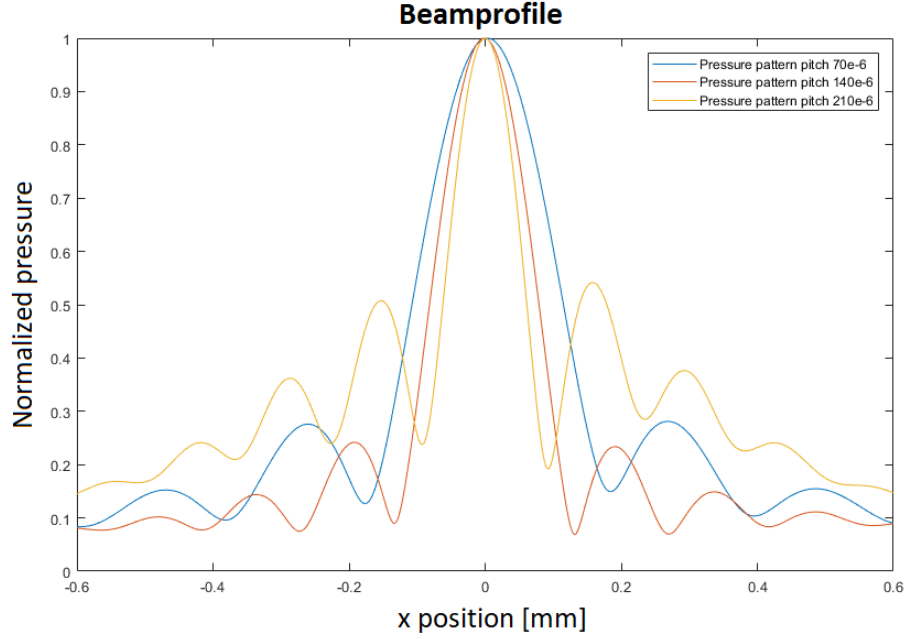


Figura 29: Comparacion de la propagacion del haz de presion para los diferentes pitches.

Ancho del haz acústico	
<i>Pitch</i>	<i>Ancho del haz(mm)(-3dB)</i>
70e-6	0.288
140e-6	0.1575
210e-6	0.1215

Tabla 8: Valores ancho del haz acústico para diferentes arrays rectangulares con *beam-forming* por columnas.

Tal y como se puede ver en la tabla y la *figura 29*, cuanto mayor es el *pitch*, más estrecho es el haz de presión, lo que nos proporciona una focalización mayor en comparación con el caso anterior en el que no se disponía de *beam-forming*. Si comparamos estos valores con el caso anterior, se puede ver que el haz de presión se reduce hasta las micras mientras que en el caso anterior se trataba de milímetros. Para ver más clara la focalización que presenta la configuración, se realiza un diagrama de radiación:

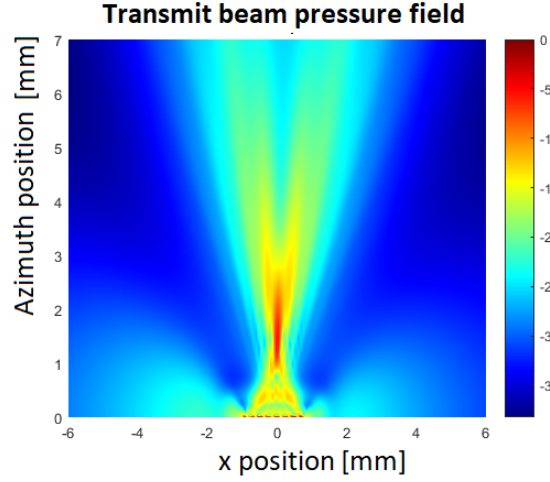


Figura 30: Diagrama de radiación del array con beam-forming por filas. Eje 'x' en función del eje 'z'.

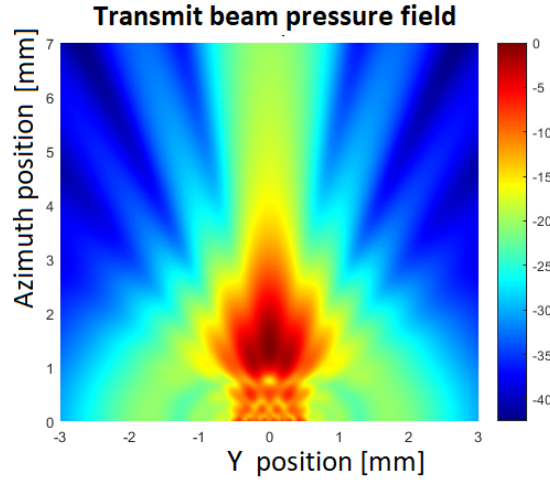


Figura 31: Diagrama de radiación lateral del array con beam-forming por filas. Eje 'y' en función del eje 'z'.

Si se analiza la figura 30 y la figura 31, se puede ver que existe un rango de focalización alrededor del milímetro hasta los dos milímetros y medio de profundidad donde la presión es mayor que en el resto de puntos. El rango de puntos donde la presión es mayor, coincide con la focalización del elemento que se encuentra en $1,5\text{mm}$, lo que nos permite corroborar que mediante *beam-forming* se puede focalizar en un punto deseado. Para llevar a cabo esta implementación, es necesario conocer los *delays* aplicados en el *beam-forming*. Para saber si esta aplicación es viable, se calculan de forma aproximada para saber en qué rango se trabaja. Si analizamos la figura 32 se puede ver que calculando la distancia del primer PMUT al punto de focalización (recta roja) es posible determinar el tiempo aproximado que es necesario para focalizar:

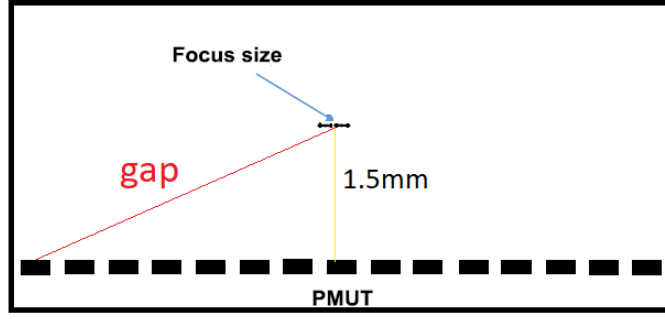


Figura 32: Esquema para calcular delays del beam-forming por filas.

$$tiempo = \frac{espacio}{velocidad} \quad (12)$$

Si analizamos, por ejemplo, el primer PMUT de la configuración de la *figura 28*, sabemos que este se encuentra a una distancia de $0,925mm$ en el eje 'x' respecto al centro. Y que la focalización se realiza a $1,5mm$.

$$espacio = \sqrt{(1,5 \cdot 10^{-3})^2 + (0,925 \cdot 10^{-3})^2} = 1,7622mm \quad (13)$$

Y, aplicando (13) en (12):

$$tiempo = \frac{0,0017622}{750} = 2,34\mu s \quad (14)$$

Si (14) se compara con los valores reales de los *delays*:

Valores de los <i>delays</i> para <i>beam-forming</i> por columnas	
<i>N de columna</i>	<i>Delay</i>
29	1.6945e-07
30	1.2971e-07
31	9.4476e-08
32	6.4196e-08
33	3.9321e-08
34	2.0256e-08
35	7.3403e-09
36	8.1832e-10
37	8.1832e-10
38	7.3403e-09
39	2.0256e-08
40	3.9321e-08
41	6.4196e-08
42	9.4476e-08
43	1.2971e-07

Tabla 9: Valores de los *delays* empleados para la implementación de arrays cuadrados con *beam-forming* por columnas.

Comparando el valor calculado con los valores implementados se puede ver que el resultado es aproximado y que por lo general hablamos de *delays* del orden de microsegundos y más pequeños.

4.2.1. Conclusiones.

A lo largo de este apartado se han analizado los efectos de la geometría de un *array* de PMUTs con *beam-forming* por filas. En particular se ha analizado:

- Máximos en el punto donde se aplica *beam-forming*.
- Diagrama de radiación en función del eje 'x' focalizando.
- Diagrama de presión del *array* de PMUTs.

Y, las conclusiones de este apartado es que se ha visto como:

- Se ha podido ver que con el *array* se mejora mucho la directividad y, al igual que en el caso anterior, cuanto más grande es el *array* más lejos se encuentra el punto de campo lejano.
- Aplicando *beam-forming* por filas se consigue resolución en el eje 'x' en un rango de valores próximos a $z = 1'5mm$ pero no el eje 'y'.

- Analizando los *dealsys*, se puede ver que la implementación sería posible pero lo ideal sería aplicar *beam-forming* a cada PMUT de forma individual.

Para mejorar la resolución y poder focalizar en un punto en concreto, se realiza el estudio de *arrays* con *beam-forming* a cada PMUT a nivel individual.

4.3. Array rectangular aplicando beam-forming a cada PMUT

A pesar de que este apartado no ha sido implementado por el grupo ECAS, resulta interesante analizarlo. En este apartado se estudia una matriz de PMUTs aplicando *beam-forming* a cada PMUT y se puede observar cómo ahora nos focalizamos en un punto a diferencia de las simulaciones anteriores donde se aplicaba *beam-forming* por filas.

En primer lugar, se analiza la presión en función del eje 'z' donde se pueden apreciar los máximos en $z = 1,5mm$ y se focalizará en el centro del *array* en campo cercano, es decir, se focalizará en ($x = 0, y = 0$ y $z = 1,4mm$).

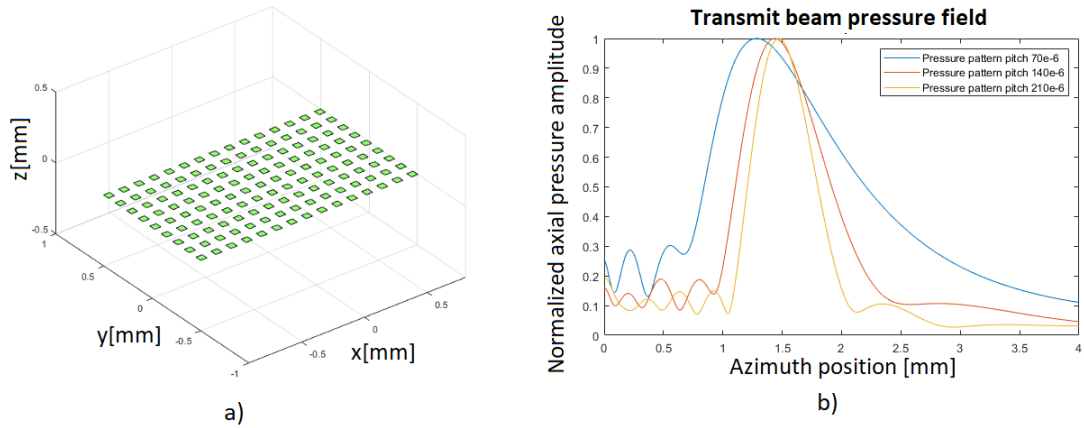


Figura 33: a) Representación de un array de 15x9 elementos. b) Comparación de presión normalizada en el eje 'z' con *beam-forming* individual.

Como en el caso anterior, cuando se realiza *beam-forming* se focaliza en campo cercano. Por ello, se analiza la presión en función del eje 'x' cuando $z = 1,4mm$ pero en este caso, se ha realizado una comparación variando la distancia entre los PMUTs. En la figura 34 se comparan los efectos que producen aumentar la distancia entre los PMUTs para tres configuraciones distintas: una distancia entre PMUTs de 70um, el doble de ésta 140um y por último para 210um.

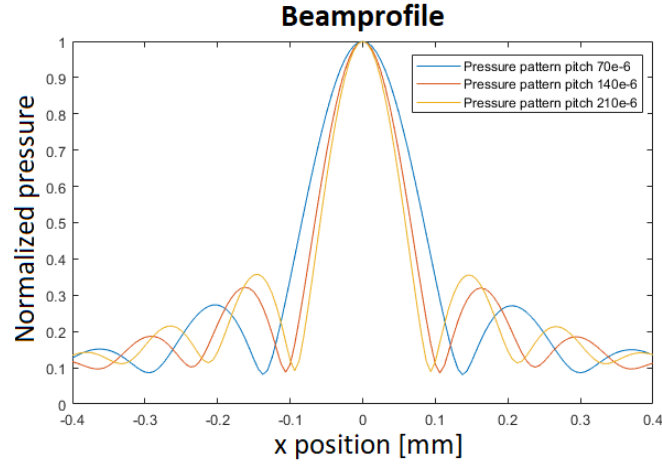


Figura 34: Comparacion de la propagacion del haz de presion para los diferentes pitches.

Al igual que en la figura 29, en la figura 34 la configuración con el *kerf* mayor presenta un haz de presión más estrecho pero en este caso, las oscilaciones laterales se encuentran bastante más alejadas del centro. Todas las configuraciones presentan una forma similar y presentan una serie que se mantiene a lo largo del eje 'x'. Por el contrario la figura 29, presenta las oscilaciones muy tempranamente, impidiendo obtener un haz tan estrecho y tan bien definido como el de esta última figura.

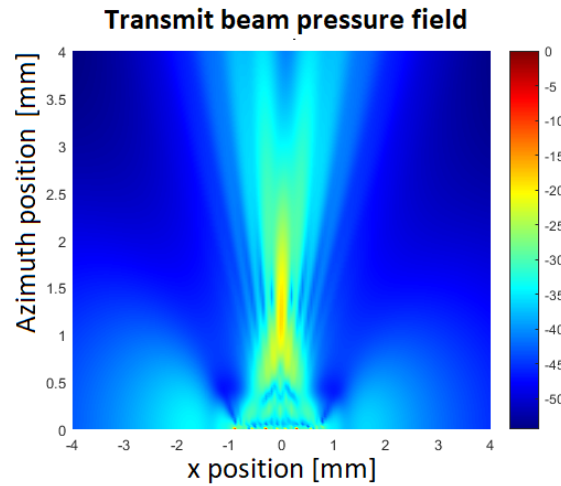


Figura 35: Diagrama de radiación del array con beam-forming a cada PMUT.

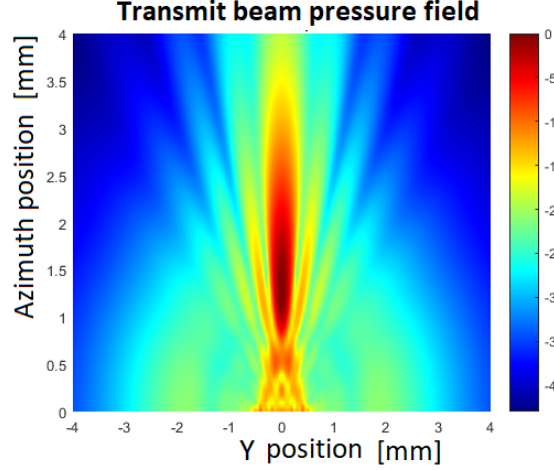


Figura 36: Diagrama de radiación lateral del array con *beam-forming* a cada PMUT.

Finalmente, analizando el diagrama de radiación se puede ver que el punto donde se ha focalizado, es donde se encuentra mayor presión, lo que nos proporcionaría una buena focalización y por ende, una buena resolución. A diferencia del caso anterior, la presión mayor se encuentra en el punto de $z = 1,4$ que es donde hemos focalizado, mientras que en la *figura 30*, la presión mayor se encontraba en un rango de puntos. Esto se debe al tipo de *beam-forming* aplicado, antes era mediante columnas y en este apartado es *beam-forming* a cada elemento. Si analizamos las *figuras 31* y *figura 36* se puede ver que la presión no se focaliza en un único punto cuando se analiza el eje 'y' lo que nos indica que no se obtiene una buena resolución lateral en ninguno de los casos. Finalmente, se puede ver como aplicar *beam-forming* a cada PMUT reduce en el ancho del haz de presión y se obtiene una buena resolución debido a la focalización.

4.3.1. Conclusiones.

A lo largo de este apartado se han analizado los efectos de la geometría de un array de PMUTs con *beam-forming* aplicado a cada PMUT. En particular se ha analizado:

- Máximos en el punto donde se aplica *beam-forming* a nivel individual.
- Diagrama de radiación en función del eje 'x' focalizando.
- Diagrama de presión del array de PMUTs.

Y, las conclusiones de este apartado es que se ha visto como:

- El diagrama de radiación que presenta focalización en un punto cuando se aplica *beam-forming* a cada PMUT.
- Aplicando *beam-forming* por filas se consigue resolución en el eje 'x' pero no el eje 'y'.

- Focalización en un punto mediante el diagrama de presión del *array* de PMUTs mediante *beam-forming* a cada PMUT.

5. Conclusión y trabajo futuro.

En este trabajo se ha realizado un estudio de las posibles geometrías de sistemas de PMUTs que se utilizan para la realización de imágenes (*imaging*) para *fingerprints* y aplicaciones biomédicas. Como ya se ha comentado a lo largo del proyecto, partiendo de los sistemas que se están implementando en el grupo ECAS se ha visto qué geometría, sistema de *beam-forming* o de qué forma hay que excitar esa geometría para obtener la máxima resolución lateral. En primer lugar, se ha obtenido como base un único dispositivo, donde se ha estudiado sus prestaciones a nivel acústico (los campos radiados así como selectividad). Se ha visto que con un único dispositivo no es posible conseguir la selectividad lateral necesaria, con lo cual, se ha optado por la implementación de *arrays* de PMUTs. Inicialmente se ha realizado un *array* lineal pero se ha visto que resultaban más interesantes los *arrays* cuadrados por las prestaciones que éstos presentaban y la utilización de *beam-forming*, tanto por columnas como a nivel de dispositivo. A lo largo del proyecto se ha visto que si se quiere obtener una buena selectividad tanto en 'x' como en 'y' la técnica óptima es realizar *beam-forming* a nivel de dispositivo, no obstante, su implementación práctica resultaba inviable por lo que se ha decidido realizar *beam-forming* a nivel de fila. Mediante la implementación de los *arrays* con *beam-forming* por filas, se ha estudiado qué efectos tiene cuando se varía el *pitch* del *beam-forming*, es decir, cuando varías las filas excitadas, viendo como la selectividad es mayor cuanto mayor es el ancho del *pitch*.

Este trabajo nos ha servido como un estudio teórico para una futura implementación práctica en el laboratorio. Como trabajo futuro y para evitar el uso de *beam-forming* a nivel de dispositivo, una implementación posible es la realización de *arrays* redondos donde se conseguiría una focalización central o, lo que es lo mismo, una focalización tanto en 'x' como en 'y'.

6. Referencias.

Referencias

- [1] P. Krause, E. Obermeier, and W. Wehl, "Backshooter - A New Smart Micromachined Single-chip Inkjet Printhead, in Solid-State Sensors and Actuators, 1995 and Eurosensors IX.. Transducers '95. The 8th International Conference on, 1995, pp. 325- 328.
- [2] .Analog Devices iMEMS."T. Scheiter, H. Kapels, K. G. Oppermann, M. Steger, C. Hierold, W. M. Werner, and H. J. Timme, "Full integration of a pressure-sensor system into a standard BiCMOS process, in EUROSENSORS XI Meeting, Warsaw, Poland, 1997, pp. 211-214.
- [3] R. Bashir, "BioMEMS: state-of-the-art in detection, opportunities and prospects, .Advanced Drug Delivery Reviews, vol. 56, pp. 1565-1586, 2004.
- [4] P. S. Waggoner and H. G. Craighead, "Micro- and nanomechanical sensors for environmental, chemical, and biological detection," Lab Chip, vol. 7, pp. 1238–1255, 2007.
- [5] Akustica, .AKU1126 Product Brief .Available on-line at <http://www.akustica.com>.
- [6] Jorgen Arendt Jensen, "field-ii.dk University of Denmark, August 1990.
- [7] [2b] K. H. L. Chau, S. R. Lewis, Y. Zhao, R. T. Howe, S. F. Bart, and R. G. Marcheselli, .An Integrated Force-balanced Capacitive Accelerometer For Low-G Applications, in Solid- State Sensors and Actuators, 1995 and Eurosensors IX.. Transducers '95. The 8th International Conference on, 1995, pp. 593-596.
- [8] Joontaek Jung, "Review of piezoelectric micromachined ultrasonic transducers and their applications," J. Micromech. Microeng. 27 113001, March 2018.
- [9] Jiang, X., Lu, Y., Tang, H., Tsai, J., Ng, E., Daneman, M., Boser, B. and Horsely, D. "Monolithic ultrasound fingerprint sensor," Microsystems and Nanoengineering 17059, November 2017.
- [10] Lu, Y., Tang, H., Fung, S., Boser, B. and Horsely, D. "Short-range and high-resolution ultrasound imaging using an 8MHz aluminum nitride pmut array," IEEE, January 2015.
- [11] Jiang, X., Lu, Y., Tang, H., Tsai, J., Ng, E., Tsai, J., Boser, B. and Horsely, D. "Ultrasonic Fingerprint Sensor With Transmit Beamforming Based on a PMUT Array Bonded to CMOS Circuitry," IEEE, Vol. 64, No9 , September 2017.
- [12] Qiu, Y., Gigliotti, J., Wallace, M., Griggio, F., Demore, C., Cochran, S. and Troler-McKinstry, T., "Arrays for Integrated Sensing, Actuation and Imaging," Sensor ISSN 1424-8220, 2015.

- [13] Lu, Y., Tang, H., Fung, S. and Boser, B. "Pulse-Echo Ultrasound Imaging Using an AlN Piezoelectric Micromachined Ultrasonic Transducer Array With Transmit Beam-Forming," IEEE, November 2015.
- [14] Lu, Y., Heidari, A., and Horsely, D. "A High Fill-Factor Annular Array of High Frequency Piezoelectric Micromachined Ultrasonic Transducers," IEEE, September 2014.

7. Anexos.

En este apartado se mostrarán los códigos implementados para las simulaciones de los PMUTs:

7.0.1. Presión en función de la profundidad en un único PMUT con forma de pistón

```
%% nico transductor.
%% make sure that field_init has been called
eval('field_init(0)', '1;')
clear;

%% Definición del transductor (tambor)
R0 = 0.025/1000;
ele_size0 = 0.0025/1000;
R1 = 0.1/1000;
ele_size1 = 0.01/1000;
R2 = 0.5/1000;
ele_size2 = 0.05/1000;
R3 = 1/1000;
ele_size3 = 0.1/1000;
R4 = 2.5/1000;
ele_size4 = 0.5/1000;

fs = 1e9;
f = 8e6;
c = 750;
lambda = c/f;

%% Se genera la apertura
emit_aperture0 = xdc_piston(R0, ele_size0);
emit_aperture1 = xdc_piston(R1, ele_size1);
emit_aperture2 = xdc_piston(R2, ele_size2);
emit_aperture3 = xdc_piston(R3, ele_size3);
emit_aperture4 = xdc_piston(R4, ele_size4);
eval('close(1)', '1;')

%% Definimos la excitación de la transmisión de apertura.

Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gausspuls(t_ir, f, Bw);
set_sampling(fs);
```

```

xdc_impulse (emit_aperture0, impulse_response);
xdc_impulse (emit_aperture1, impulse_response);
xdc_impulse (emit_aperture2, impulse_response);
xdc_impulse (emit_aperture3, impulse_response);
xdc_impulse (emit_aperture4, impulse_response);

%Excitaci n
ex_periods = 2;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture0, excitation);
xdc_excitation (emit_aperture1, excitation);
xdc_excitation (emit_aperture2, excitation);
xdc_excitation (emit_aperture3, excitation);
xdc_excitation (emit_aperture4, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepthStart = 0/1000;
measDepthEnd = 20/1000;
x = 0/1000;
Nmp = 1601;

mx = x*ones(Nmp,1);
my = zeros(Nmp, 1);
mz = linspace(measDepthStart,measDepthEnd,Nmp)';
measurement_points = [mx,my,mz];

% Plot para la Figura 1
figure(1);
hold on;
plot3(measurement_points(:,1)*1000,measurement_points(:,2)*1000,
      measurement_points(:,3)*1000, 'o', 'linewidth', 6)
axis tight

%% CALCULATE TRANSMIT PRESSURE

%Transmit Pressure
[pressure_tx0, startTime_tx0] = calc_hp(emit_aperture0, measurement_points);
[pressure_tx1, startTime_tx1] = calc_hp(emit_aperture1, measurement_points);
[pressure_tx2, startTime_tx2] = calc_hp(emit_aperture2, measurement_points);
[pressure_tx3, startTime_tx3] = calc_hp(emit_aperture3, measurement_points);
[pressure_tx4, startTime_tx4] = calc_hp(emit_aperture4, measurement_points);

%% PLOT BP

% figure();
% subplot(2,1,1);

```



```

bpx0 = sqrt(mean(pressure_tx0.^2));
% bp = reshape(bp, Nmp, Nmpx);
bpx0=bpx0/max(bpx0);
bpx0 = 20*log10(bpx0);
% plot(mz*1000, bpx0);
% title(sprintf('Transmit beam pressure field'))
% xlabel('Azimuth position [mm]');
% ylabel('Normalized axial pressure amplitdue');

% subplot(2,1,2)
bpx1 = sqrt(mean(pressure_tx1.^2));
% bp = reshape(bp, Nmp, Nmpx);
bpx1=bpx1/max(bpx1);
bpx1 = 20*log10(bpx1);
% plot(mz*1000, bpx1);
% title(sprintf('Transmit beam pressure field'))
% xlabel('Azimuth position [mm]');
% ylabel('Normalized axial pressure amplitdue');

% figure();
bpx2 = sqrt(mean(pressure_tx2.^2));
% bp = reshape(bp, Nmp, Nmpx);
bpx2 = bpx2/max(bpx2);
bpx2 = 20*log10(bpx2);
% plot(mz*1000, bpx2);
% title(sprintf('Transmit beam pressure field'))
% xlabel('Azimuth position [mm]');
% ylabel('Normalized axial pressure amplitdue');

% figure();
bpx3 = sqrt(mean(pressure_tx3.^2));
% bp = reshape(bp, Nmp, Nmpx);
bpx3 = bpx3/max(bpx3);
bpx3 = 20*log10(bpx3);
% plot(mz*1000, bpx3);
% title(sprintf('Transmit beam pressure field'))
% xlabel('Azimuth position [mm]');
% ylabel('Normalized axial pressure amplitdue');

% figure();
% bpx4 = sqrt(mean(pressure_tx4.^2));
% % bp = reshape(bp, Nmp, Nmpx);
% bpx4 = bpx4/max(bpx4);
% plot(mz*1000, bpx4);
% title(sprintf('Transmit beam pressure field'))
% xlabel('Azimuth position [mm]');

```

```

% ylabel('Normalized axial pressure amplitdue');

%%PLOT ALL TRANSMIT BEAM PRESSURE FIELD LINES

figure();
% plot(mz*1000, bpx0);
% hold on;
% plot(mz*1000, bpx1);
% hold on;
% plot(mz*1000, bpx2);
% hold on;
% plot(mz*1000, bpx3);
title(sprintf('Transmit_beam_pressure_field'))
xlabel('Azimuth_position_[mm]');
ylabel('Normalized_axial_pressure_amplitdue');

subplot(2,1,1);
plot(mz*1000, bpx2);
title(sprintf('Transmit_beam_pressure_field'))
xlabel('Azimuth_position_[mm]');
ylabel('Normalized_axial_pressure_amplitdue_(dB)');

subplot(2,1,2);
plot(mz*1000, bpx3);
title(sprintf('Transmit_beam_pressure_field'))
xlabel('Azimuth_position_[mm]');
ylabel('Normalized_axial_pressure_amplitdue_(dB)');

```

7.0.2. Presión en función de la frecuencia en un único PMUT con forma de pistón:

```

%%make sure that field_init has been called
eval('field_init(0)','1;')
clear;

%% Definición del transductor (tambor)
R=0.1/1000;
ele_size=0.01/1000;
fs = 1e9;
f0 = 20e6;
f1 = 100e6;
f2 = 200e6;
f3 = 400e6;
c = 750;
lambda = c/f0;
%% Se genera la apertura

```

```

emit_aperture0 = xdc_piston(R, ele_size);
emit_aperture1 = xdc_piston(R, ele_size);
emit_aperture2 = xdc_piston(R, ele_size);
emit_aperture3 = xdc_piston(R, ele_size);
eval('close(1)', '1;')
figure()
grid on;
show_xdc_geir(emit_aperture0, 1);
axis equal; view(3)
h_txAp = gcf;

%%Definimos la excitaci n de la transmisi n de apertura.
%Impulso_0
t_ir0 = -2/f0:1/fs:2/f0;
Bw = 0.4;
impulse_response0 = gauspuls(t_ir0, f0, Bw);
set_sampling(fs);
xdc_impulse (emit_aperture0, impulse_response0);
%Impulso_1
t_ir1 = -2/f1:1/fs:2/f1;
Bw = 0.4;
impulse_response1 = gauspuls(t_ir1, f1, Bw);
set_sampling(fs);
xdc_impulse (emit_aperture1, impulse_response1);
%Impulso_2
t_ir2 = -2/f2:1/fs:2/f2;
Bw = 0.4;
impulse_response2 = gauspuls(t_ir2, f2, Bw);
set_sampling(fs);
xdc_impulse (emit_aperture2, impulse_response2);
%Impulso_3
t_ir3 = -2/f3:1/fs:2/f3;
Bw = 0.4;
impulse_response3 = gauspuls(t_ir3, f3, Bw);
set_sampling(fs);
xdc_impulse (emit_aperture3, impulse_response3);

%Excitaci n_0
ex_periods = 2;
t_ex0 = (0:1/fs:ex_periods/f0);
excitation0 = square(2*pi*f0*t_ex0);
xdc_excitation (emit_aperture0, excitation0);
%Excitaci n_1
ex_periods = 2;
t_ex1 = (0:1/fs:ex_periods/f1);
excitation1 = square(2*pi*f1*t_ex1);
xdc_excitation (emit_aperture1, excitation1);

```

```

%Excitaci n_2
ex_periods = 2;
t_ex2 = (0:1/fs:ex_periods/f2);
excitation2 = square(2*pi*f2*t_ex2);
xdc_excitation (emit_aperture2, excitation2);
%Excitaci n_3
ex_periods = 2;
t_ex3 = (0:1/fs:ex_periods/f3);
excitation3 = square(2*pi*f3*t_ex3);
xdc_excitation (emit_aperture3, excitation3);

figure();
plot(t_ex0*1e6, excitation0);
hold on;
plot(t_ex1*1e6, excitation1);
hold on;
plot(t_ex2*1e6, excitation2);
hold on;
plot(t_ex3*1e6, excitation3);
ylim([0 1]);
grid on;
title('Excitation');
xlabel('Time_\mus');
ylabel('Amplitude');

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepth = 7/1000;
xStart = -3/1000;
xEnd = 3/1000;
Nmp = 1601;
mx = linspace(xStart,xEnd,Nmp)';
my = zeros(Nmp, 1);
mz = measDepth*ones(Nmp,1);
measurement_points = [mx,my,mz];

figure(1);
hold on;
plot3(measurement_points(:,1)*1000,measurement_points(:,2)*1000,
      measurement_points(:,3)*1000, 'o', 'linewidth', 6)
axis tight
%% CALCULATE TRANSMIT PRESSURE

%Transmit Pressure 0
[pressure_tx0, startTime_tx0] = calc_hp(emit_aperture0, measurement_points);
bpx0= sqrt(mean(pressure_tx0.^2));
bpx0=bpx0/max(bpx0);
%Transmit Pressure 1

```

```

[pressure_tx1, startTime_tx1] = calc_hp(emit_aperture1, measurement_points);
bpx1 = sqrt(mean(pressure_tx1.^2));
bpx1=bpx1/max(bpx1);
%Transmit Pressure 2
[pressure_tx2, startTime_tx2] = calc_hp(emit_aperture2, measurement_points);
bpx2 = sqrt(mean(pressure_tx2.^2));
bpx2=bpx2/max(bpx2);
%Transmit Pressure 3
[pressure_tx3, startTime_tx3] = calc_hp(emit_aperture3, measurement_points);
bpx3 = sqrt(mean(pressure_tx3.^2));
bpx3=bpx3/max(bpx3);

figure();
plot(mx*1000, bpx0);
hold on;
plot(mx*1000, bpx1);
hold on;
plot(mx*1000, bpx2);
hold on;
plot(mx*1000, bpx3);
title(sprintf('Beamprofile_diameter=_200 m _transducer_at_depth=_8mm_with_fs
=1GHz', measDepth*1000))
xlabel('Azimuth_position_[mm]');
ylabel('Normalized_pressure');

```

7.0.3. Presión en función de la diámetro en un único PMUT con forma de pistón y la directividad:

```

%%make sure that field_init has been called
eval('field_init(0)','1;')
clear;

%% Definición del transductor (tambor)
R0 = 0.025/1000;
ele_size0 = 0.0025/1000;
R1 = 0.1/1000;
ele_size1 = 0.01/1000;
R2 = 0.5/1000;
ele_size2 = 0.05/1000;
R3 = 1/1000;
ele_size3 = 0.1/1000;
fs = 1e9;
f = 8e6;
c = 750;
lambda = c/f;

```

```

%% Se genera la apertura
emit_aperture0 = xdc_piston(R0, ele_size0);
emit_aperture1 = xdc_piston(R1, ele_size1);
emit_aperture2 = xdc_piston(R2, ele_size2);
emit_aperture3 = xdc_piston(R3, ele_size3);
eval('close(1)','1;')
figure()
grid on;
show_xdc_geir(emit_aperture0, 1);
axis equal; view(3)
h_txAp = gcf;

%% Definimos la excitaci n de la transmisi n de apertura.
%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture0, impulse_response);
xdc_impulse (emit_aperture1, impulse_response);
xdc_impulse (emit_aperture2, impulse_response);
xdc_impulse (emit_aperture3, impulse_response);

%Excitaci n
% ex_periods = 2;
% t_ex=(0:1/fs:ex_periods/f);
% excitation=square(2*pi*f*t_ex);
% xdc_excitation (emit_aperture0, excitation);
% xdc_excitation (emit_aperture1, excitation);
% xdc_excitation (emit_aperture2, excitation);
% xdc_excitation (emit_aperture3, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepth = 7/1000;
x_Start = -20/1000;
x_End = 20/1000;
Nmp = 1601;
mx = linspace(x_Start,x_End,Nmp)';
my = zeros(Nmp, 1);
mz = measDepth*ones(Nmp,1);
measurement_points = [mx,my,mz];

%% Para calcular la directividad:
% % DEFINE MEASUREMENT POINTS a lo largo de una linea
% measDepth = 7/1000;
% t_ir = 0:0.001:1;

```

```

% mz = sin(2*pi*t_ir)';
% mx = cos(2*pi*t_ir)';
% my = zeros(1001, 1);
% measurement_points = [mx, my, mz];

% Plot para la Figura 1
figure(1);
hold on;
plot3(measurement_points(:,1)*1000, measurement_points(:,2)*1000, measurement_points(:,3)*1000, 'o', 'linewidth', 6)
axis tight

%% CALCULATE TRANSMIT PRESSURE
% Transmit Pressure
[pressure_tx0, startTime_tx0] = calc_hp(emit_aperture0, measurement_points);
[pressure_tx1, startTime_tx1] = calc_hp(emit_aperture1, measurement_points);
[pressure_tx2, startTime_tx2] = calc_hp(emit_aperture2, measurement_points);
[pressure_tx3, startTime_tx3] = calc_hp(emit_aperture3, measurement_points);

%% Definimos la excitaci3n de la transmisi3n de apertura.

% Excitaci3n_0
ex_periods = 2;
t_ex = (0:1/fs:ex_periods/f);
excitation0 = square(2*pi*f*t_ex);
xdc_excitation(emit_aperture0, excitation0);
% Excitaci3n_1
ex_periods = 2;
t_ex = (0:1/fs:ex_periods/f);
excitation1 = square(2*pi*f*t_ex);
xdc_excitation(emit_aperture1, excitation1);
% Excitaci3n_2
ex_periods = 2;
t_ex = (0:1/fs:ex_periods/f);
excitation2 = square(2*pi*f*t_ex);
xdc_excitation(emit_aperture2, excitation2);
% Excitaci3n_3
ex_periods = 2;
t_ex = (0:1/fs:ex_periods/f);
excitation3 = square(2*pi*f*t_ex);
xdc_excitation(emit_aperture3, excitation3);

%% CALCULATE TRANSMIT PRESSURE
% Transmit Pressure 0
[pressure_tx0, startTime_tx0] = calc_hp(emit_aperture0, measurement_points);
bpx0 = sqrt(mean(pressure_tx0.^2));

```

```

bpx0=bpx0/max(bpx0);
%Transmit Pressure 1
[pressure_tx1, startTime_tx1] = calc_hp(emit_aperture1, measurement_points);
bpx1 = sqrt(mean(pressure_tx1.^2));
bpx1=bpx1/max(bpx1);
%Transmit Pressure 2
[pressure_tx2, startTime_tx2] = calc_hp(emit_aperture2, measurement_points);
bpx2 = sqrt(mean(pressure_tx2.^2));
bpx2=bpx2/max(bpx2);
%Transmit Pressure 3
[pressure_tx3, startTime_tx3] = calc_hp(emit_aperture3, measurement_points);
bpx3 = sqrt(mean(pressure_tx3.^2));
bpx3=bpx3/max(bpx3);

figure ();
plot(mx*1000, bpx0);
hold on;
plot(mx*1000, bpx1);
hold on;
plot(mx*1000, bpx2);
hold on;
plot(mx*1000, bpx3);
title(sprintf('Beamprofile_at_depth_=%0.3gmm', measDepth*1000))
xlabel('x_position_[mm]');
ylabel('Presi n_normalizada');

%% Para calcular la directividad
% figure();
% polar(angulo, bpx0)
% hold on;
% polar(angulo, bpx1);
% hold on;
% polar(angulo, bpx2);
% hold on;
% polar(angulo, bpx3);
% hold on;
% title(sprintf('Beamprofile at depth = %0.3gmm', measDepth*1000))

```

7.0.4. Presión en función del eje 'z' en único array PMUT lineal

```

%% make sure that field_init has been called
eval('field_init(0)', '1;')
clear;

%% DEFINE PARAMETERS
c = 750;

```



```

f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x = 73e-6;
kerf_y = 73e-6;
no_ele_x_1 = 7;
no_ele_y_1 = 1;
%no_ele_x_2 = 5;
%no_ele_y_2 = 1;
%
%no_ele_x_3 = 9;
%no_ele_y_3 = 1;
no_sub_x = 5;
no_sub_y = 5;
focus = [0 0 1.5]/100;

%% FIND WHICH ELEMENTS TO USE
enabled_1 = ones(no_ele_x_1, no_ele_y_1);
%enabled_2 = ones(no_ele_x_2, no_ele_y_2);
%enabled_3 = ones(no_ele_x_3, no_ele_y_3);

%% DEFINE THE TRANSDUCER
emit_aperture_1 = xdc_2d_array (no_ele_x_1, no_ele_y_1, height, width, kerf_x,
    kerf_y, enabled_1, no_sub_x, no_sub_y, focus);
%emit_aperture_2 = xdc_2d_array (no_ele_x_2, no_ele_y_2, height, width, kerf_x,
    kerf_y, enabled_2, no_sub_x, no_sub_y, focus);
%emit_aperture_3 = xdc_2d_array (no_ele_x_3, no_ele_y_3, height, width, kerf_x,
    kerf_y, enabled_3, no_sub_x, no_sub_y, focus);

%% DELAYS
delay_times_1 = zeros(1, no_ele_x_1*no_ele_y_1);
xdc_focus_times(emit_aperture_1, 0, delay_times_1);
data = xdc_get(emit_aperture_1, 'focus');
%delay_times_2 = zeros(1, no_ele_x_2*no_ele_y_2);
%xdc_focus_times(emit_aperture_2, 0, delay_times_2);
%data = xdc_get(emit_aperture_2, 'focus');
%delay_times_3 = zeros(1, no_ele_x_3*no_ele_y_3);
%xdc_focus_times(emit_aperture_3, 0, delay_times_3);
%data = xdc_get(emit_aperture_3, 'focus');

%% Definimos la excitaci3n de la transmisi3n de apertura.
%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir, f, Bw);

```

```

set_sampling(fs);
xdc_impulse (emit_aperture_1, impulse_response);
% xdc_impulse (emit_aperture_2, impulse_response);
% xdc_impulse (emit_aperture_3, impulse_response);

%Excitaci n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture_1, excitation);
% xdc_excitation (emit_aperture_2, excitation);
% xdc_excitation (emit_aperture_3, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepthStart = 0/1000;
measDepthEnd = 50/1000;
x = 0/1000;
Nmp = 1000;
mx = x*ones(Nmp,1);
my = zeros(Nmp, 1);
mz = linspace(measDepthStart, measDepthEnd, Nmp)';
measurement_points = [mx,my,mz];

%% Do the Calculation with Beamforming
%Transmit Pressure
[pressure_tx_1, startTime_tx_1] = calc_hp(emit_aperture_1, measurement_points);
% [pressure_tx_2, startTime_tx_2] = calc_hp(emit_aperture_2, measurement_points)
;
% [pressure_tx_3, startTime_tx_3] = calc_hp(emit_aperture_3, measurement_points)
;

%% PLOTS
%% %% Ploteamos las columnas excitadas
figure()
grid on;
show_xdc_geir(emit_aperture_1, 1);
axis equal; view(3)
h_txAp = gcf;
%
% figure()
% grid on;
% show_xdc_geir(emit_aperture_2, 1);
% axis equal; view(3)
% h_txAp = gcf;
%
% figure()
% grid on;

```

```

% show_xdc_geir(emit_aperture_3, 1);
% axis equal; view(3)
% h_txAp = gcf;

figure ()
bpx_1 = sqrt(mean(pressure_tx_1.^2));
bpx_1 = bpx_1/max(bpx_1);
% bpx_2 = sqrt(mean(pressure_tx_2.^2));
% bpx_2 = bpx_2/max(bpx_2);
% bpx_3 = sqrt(mean(pressure_tx_3.^2));
% bpx_3 = bpx_3/max(bpx_3);

plot(mz*1000, bpx_1)
% hold on;
% plot(mz*1000, bpx_2)
% hold on;
% plot(mz*1000, bpx_3)
title(sprintf('Transmit_beam_pressure_field '))
xlabel('Azimuth_position_[mm]');
ylabel('Normalized_axial_pressure_amplitdue');

```

7.0.5. Presión en función de la diámetro y la directividad en un array PMUT lineal:

```

%% make sure that field_init has been called
eval('field_init(0)', '1;')
clear;

%% DEFINE PARAMETERS
c = 750;
f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x = 70e-6;
kerf_y = 70e-6;
no_ele_x_1 = 7;
no_ele_y_1 = 1;
no_ele_x_2 = 5;
no_ele_y_2 = 1;
no_ele_x_3 = 9;
no_ele_y_3 = 1;
no_sub_x = 5;
no_sub_y = 5;

```

```

focus_1 = [0 0 1.5]/1000;
% focus_2 = [0 0 1.5]/1000;
% focus_3 = [0 0 1.5]/1000;

%% FIND WHICH ELEMENTS TO USE
enabled_1 = ones(no_ele_x_1, no_ele_y_1);
enabled_2 = ones(no_ele_x_2, no_ele_y_2);
enabled_3 = ones(no_ele_x_3, no_ele_y_3);

%% DEFINE THE TRANSDUCER
emit_aperture_1 = xdc_linear_array (no_ele_x_1, width, height, kerf_x, no_sub_x,
no_sub_y, focus_1);
emit_aperture_2 = xdc_linear_array (no_ele_x_2, width, height, kerf_x, no_sub_x,
no_sub_y, focus_1);
emit_aperture_3 = xdc_linear_array (no_ele_x_3, width, height, kerf_x, no_sub_x,
no_sub_y, focus_1);

%% DELAYS
delay_times_1 = zeros(1,no_ele_x_1*no_ele_y_1);
xdc_focus_times(emit_aperture_1, 0, delay_times_1);
data = xdc_get(emit_aperture_1, 'focus');

delay_times_2 = zeros(1,no_ele_x_2*no_ele_y_2);
xdc_focus_times(emit_aperture_2, 0, delay_times_2);
data = xdc_get(emit_aperture_2, 'focus');

delay_times_3 = zeros(1,no_ele_x_3*no_ele_y_3);
xdc_focus_times(emit_aperture_3, 0, delay_times_3);
data = xdc_get(emit_aperture_3, 'focus');

%% Definimos la excitaci n de la transmisi n de apertura.
%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture_1, impulse_response);
xdc_impulse (emit_aperture_2, impulse_response);
xdc_impulse (emit_aperture_3, impulse_response);

%Excitaci n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture_1, excitation);
xdc_excitation (emit_aperture_2, excitation);
xdc_excitation (emit_aperture_3, excitation);

```

```

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepth_1 = 1.5/1000;
measDepth_2 = 1.5/1000;
measDepth_3 = 1.5/1000;
x_Start = -8/1000;
x_End = 8/1000;
Nmp = 1601;
mx = linspace(x_Start, x_End, Nmp)';
my = zeros(Nmp, 1);
mz_1 = measDepth_1*ones(Nmp, 1);
mz_2 = measDepth_2*ones(Nmp, 1);
mz_3 = measDepth_3*ones(Nmp, 1);
measurement_points_1 = [mx,my,mz_1];
measurement_points_2 = [mx,my,mz_2];
measurement_points_3 = [mx,my,mz_3];

%% Para la directividad:
% measDepth_1 = 1.5/1000;
% measDepth_2 = 1.5/1000;
% measDepth_3 = 1.5/1000;
% t_ir = 0:0.001:1;
% mz = sin(2*pi*t_ir)';
% mx = cos(2*pi*t_ir)';
% my = zeros(1001, 1);
% measurement_points = [mx,my,mz];

%% Do the Calculation with Beamforming
% Transmit Pressure
[pressure_tx_1, startTime_tx_1] = calc_hp(emit_aperture_1, measurement_points_1);
[pressure_tx_2, startTime_tx_2] = calc_hp(emit_aperture_2, measurement_points_2);
[pressure_tx_3, startTime_tx_3] = calc_hp(emit_aperture_3, measurement_points_3);

%% PLOTS
%% %% Ploteamos las columnas excitadas
% figure()
% grid on;
% show_xdc_geir(emit_aperture, 1);
% axis equal; view(3)
% h_txAp = gcf;

figure()
bpx_1 = sqrt(mean(pressure_tx_1.^2));
bpx_1 = bpx_1/max(bpx_1);

```

```

bpx_2 = sqrt(mean(pressure_tx_2.^2));
bpx_2 = bpx_2/max(bpx_2);
bpx_3 = sqrt(mean(pressure_tx_3.^2));
bpx_3 = bpx_3/max(bpx_3);
plot(mx*1000, bpx_1)
hold on;
plot(mx*1000, bpx_2)
hold on;
plot(mx*1000, bpx_3)
%title(sprintf('Beamprofile at depth = %0.3gmm', measDepth*1000))
xlabel('x_position_[mm]');
ylabel('Presi n_normalizada_');

%%Para la directividad:
%figure ()
%bpx_1 = sqrt(mean(pressure_tx_1.^2));
%bpx_1 = bpx_1/max(bpx_1);
%bpx_2 = sqrt(mean(pressure_tx_2.^2));
%bpx_2 = bpx_2/max(bpx_2);
%bpx_3 = sqrt(mean(pressure_tx_3.^2));
%bpx_3 = bpx_3/max(bpx_3);
%angulo = 0:0.002*pi:2*pi;
%polar(angulo, bpx_1)
%hold on;
%polar(angulo, bpx_2)
%hold on;
%polar(angulo, bpx_3)
%title(sprintf('Beamprofile at depth = %0.3gmm', measDepth*1000))

```

7.0.6. Presión en en función de la profundidad en array rectangular sin beam-forming.

```

%% make sure that field_init has been called
eval('field_init(0)', '1;')
clear;

%% DEFINE PARAMETERS
c = 750;
f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x = 73e-6;
kerf_y = 73e-6;

```

```

no_ele_x = 72;
no_ele_y = 9;
no_sub_x = 5;
no_sub_y = 5;
focus = [0 0 1.5]/100;

%% FIND WHICH ELEMENTS TO USE
enabled = ones(no_ele_x, no_ele_y);

%% DEFINE THE TRANSDUCER
emit_aperture= xdc_2d_array (no_ele_x, no_ele_y, height, width, kerf_x, kerf_y,
    enabled, no_sub_x, no_sub_y, focus);

%% DELAYS
delay_times = zeros(1,no_ele_x*no_ele_y);
xdc_focus_times(emit_aperture, 0, delay_times);
data = xdc_get(emit_aperture, 'focus');

%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture, impulse_response);

%Excitaci n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepthStart = 0/1000;
measDepthEnd = 50/1000;
x = 0/1000;
Nmp = 1000;
mx = x*ones(Nmp,1);
my = zeros(Nmp, 1);
mz = linspace(measDepthStart,measDepthEnd,Nmp)';
measurement_points = [mx,my,mz];

%% Do the Calculation with Beamforming
%Transmit Pressure
[pressure_tx, startTime_tx] = calc_hp(emit_aperture, measurement_points);

%% PLOTS
%% % Ploteamos las columnas excitadas

```

```

figure()
grid on;
show_xdc_geir(emit_aperture, 1);
axis equal; view(3)
h_txAp = gcf;

figure ()
bpx = sqrt(mean(pressure_tx.^2));
bpx = bpx/max(bpx);
plot(mz*1000, bpx)
title(sprintf('Transmit_beam_pressure_field'))
xlabel('Azimuth_position_[mm]');
ylabel('Normalized_axial_pressure_amplitdue');

```

7.0.7. Presión en un array rectangular sin beam-forming y diagrama de presión.

```

%% make sure that field_init has been called
eval('field_init(0)','1;')
clear;

%% DEFINE PARAMETERS
c = 750;
f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x = 70e-6;
kerf_y = 70e-6;
no_ele_x = 72;
no_ele_y = 9;
no_sub_x = 10;
no_sub_y = 10;
focus = [0 0 1]/1000;

%% FIND WHICH ELEMENTS TO USE
enabled = ones(no_ele_x, no_ele_y);

%% DEFINE THE TRANSDUCER
emit_aperture= xdc_2d_array (no_ele_x, no_ele_y, height, width, kerf_x, kerf_y,
    enabled, no_sub_x, no_sub_y, focus);

%% DELAYS
delay_times = zeros(1,no_ele_x*no_ele_y);

```



```

xdc_focus_times(emit_aperture, 0, delay_times);
data = xdc_get(emit_aperture, 'focus');

%% Definimos la excitaci n de la transmisi n de apertura.

%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture, impulse_response);

%Excitaci n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea

% measDepth = 0.1/1000;
% x_Start = -8/1000;
% x_End = 8/1000;
% Nmp = 1601;
% mx = linspace(x_Start, x_End, Nmp)';
% my = zeros(Nmp, 1);
% mz = measDepth*ones(Nmp, 1);
%
% measurement_points = [mx,my,mz];

% PARA EL COLORMAP
zStart = 0/1000;
zEnd = 7/1000;
xStart = -6/1000;
xEnd = 6/1000;
Nmpx = 150;
Nmpz = 150;
mx = linspace(xStart,xEnd,Nmpx)';
my = zeros(Nmpx*Nmpz, 1);
mz = linspace(zStart,zEnd,Nmpz)';
[X,Z] = meshgrid(mx,mz);
measurement_points = [X(:),my,Z(:)];

%% Do the Calculation with Beamforming
%Transmit Pressure
[pressure_tx, startTime_tx] = calc_hp(emit_aperture, measurement_points);

```

```

%% PLOTS
% figure ( )
% bpx = sqrt(mean(pressure_tx.^2));
% bpx = bpx/max(bpx);
% plot(mx*1000, bpx)
% title(sprintf('Beamprofile at depth =%0.3gmm', measDepth*1000))
% xlabel('x position [mm]');
% ylabel('Presi n normalizada ');

%Para el colormap
figure()
bp = sqrt(mean(pressure_tx.^2));
bp = reshape(bp, Nmpz, Nmpx);
bp = bp/max(bp(:));
pcolor(mx*1000,mz*1000, 20*log10(bp));
shading interp
title(sprintf('Transmit_beam_pressure_field'))
xlabel('Azimuth [mm]');
ylabel('Range [mm]');
%axis([-25 0]); % Set dynamic range
colormap(jet(256));
colorbar

```

7.0.8. Presión en en función de la profundidad en array rectangular con beam-forming por filas.

```

%% Comparaci n entre diferentes pitch(kerf's)
%% make sure that field_init has been called
eval('field_init(0)','1;')
clear;

%% DEFINE PARAMETERS
c = 750;
f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x_1 = 70e-6;
kerf_y_1 = 70e-6;
kerf_x_2 = 140e-6;
kerf_y_2 = 140e-6;
kerf_x_3 = 210e-6;

```

```

kerf_y_3 = 210e-6;
no_ele_x = 72;
no_ele_y = 9;
no_sub_x = 10;
no_sub_y = 10;
N_columns_enables = 15;
focus = [0 0 1.5]/1000;

%% FIND WHICH ELEMENTS TO USE
enabled=zeros(no_ele_x, no_ele_y);
enabled(29,:)=ones(1,no_ele_y);
enabled(30,:)=ones(1,no_ele_y);
enabled(31,:)=ones(1,no_ele_y);
enabled(32,:)=ones(1,no_ele_y);
enabled(33,:)=ones(1,no_ele_y);
enabled(34,:)=ones(1,no_ele_y);
enabled(35,:)=ones(1,no_ele_y);
enabled(36,:)=ones(1,no_ele_y);
enabled(37,:)=ones(1,no_ele_y);
enabled(38,:)=ones(1,no_ele_y);
enabled(39,:)=ones(1,no_ele_y);
enabled(40,:)=ones(1,no_ele_y);
enabled(41,:)=ones(1,no_ele_y);
enabled(42,:)=ones(1,no_ele_y);
enabled(43,:)=ones(1,no_ele_y);

%% DEFINE THE TRANSDUCER
emit_aperture_1 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_1,
    kerf_y_1, enabled, 1, 1, focus);
emit_aperture_2 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_2,
    kerf_y_2, enabled, 1, 1, focus);
emit_aperture_3 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_3,
    kerf_y_3, enabled, 1, 1, focus);

%% DELAYS
delay_times = zeros(1,N_columns_enables*no_ele_y);
xdc_focus_times(emit_aperture, 0, delay_times);
data = xdc_get(emit_aperture, 'focus');
%% Delaysrow_1
delaysrow_1 = [-1.6945e-07 -1.2971e-07 -9.4476e-08 -6.4196e-08 -3.9321e-08 -2.0256e
    -08 -7.3403e-09 -8.1832e-10 -8.1832e-10 -7.3403e-09 -2.0256e-08 -3.9321e-08
    -6.4196e-08 -9.4476e-08 -1.2971e-07];

%delaysrow = [-2.3356e-8 0 -2.3356e-8];
delay_times_1 = 1:no_ele_y*N_columns_enables;
m = 1;
while m <= no_ele_y*N_columns_enables

```

```

    for n = 1:N_columns_enables
        delay_times_1(m)= delaysrow_1(n);
        m = m+1;
    end
end

%% Delaysrow_2
delaysrow_2 = [-3.6945e-07 -2.8765e-07 -2.1306e-07 -1.4711e-07 -9.1428e-08 -4.7670e-08
-1.7425e-08 -1.9515e-09 -1.9515e-09 -1.7425e-08 -4.7670e-08 -9.1428e-08
-1.4711e-07 -2.1306e-07 -2.8765e-07];

%delaysrow = [-2.3356e-8 0 -2.3356e-8];
delay_times_2 = 1:no_ele_y*N_columns_enables;
m = 1;
while m <= no_ele_y*N_columns_enables
    for n = 1:N_columns_enables
        delay_times_2(m)= delaysrow_2(n);
        m = m+1;
    end
end

%% Delaysrow_3
delaysrow_3 = [-6.2349e-07 -4.9329e-07 -3.7169e-07 -2.6126e-07 -1.6522e-07 -8.7518e-08
-3.2383e-08 -3.6511e-09 -3.6511e-09 -3.2383e-08 -8.7518e-08 -1.6522e-07
-2.6126e-07 -3.7169e-07 -4.9329e-07];

%delaysrow = [-2.3356e-8 0 -2.3356e-8];
delay_times_3 = 1:no_ele_y*N_columns_enables;
m = 1;
while m <= no_ele_y*N_columns_enables
    for n = 1:N_columns_enables
        delay_times_3(m)= delaysrow_3(n);
        m = m+1;
    end
end

%% Procedure for creating a focus time line for an aperture
xdc_focus_times(emit_aperture_1, 0, delay_times_1);
data = xdc_get(emit_aperture_1, 'focus');

xdc_focus_times(emit_aperture_2, 0, delay_times_2);
data = xdc_get(emit_aperture_2, 'focus');

xdc_focus_times(emit_aperture_3, 0, delay_times_3);
data = xdc_get(emit_aperture_3, 'focus');

```

```

%% Definimos la excitaci n de la transmisi n de apertura.

%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gausspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture_1, impulse_response);
xdc_impulse (emit_aperture_2, impulse_response);
    xdc_impulse (emit_aperture_3, impulse_response);

%Excitaci n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture_1, excitation);
xdc_excitation (emit_aperture_2, excitation);
xdc_excitation (emit_aperture_3, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepthStart = 0/1000;
measDepthEnd = 8/1000;
x = 0/1000;
Nmp = 1000;
mx = x*ones(Nmp,1);
my = zeros(Nmp, 1);
mz = linspace(measDepthStart,measDepthEnd,Nmp)';
measurement_points = [mx,my,mz];

%%Do the Calculation with Beamforming
%Transmit Pressure_1
[pressure_tx_1, startTime_tx_1] = calc_hp(emit_aperture_1, measurement_points);
%Transmit Pressure_2
[pressure_tx_2, startTime_tx_2] = calc_hp(emit_aperture_2, measurement_points);
%Transmit Pressure_3
[pressure_tx_3, startTime_tx_3] = calc_hp(emit_aperture_3, measurement_points);

%% PLOTS
%% Ploteamos las columnas excitadas
figure()
grid on;
show_xdc_geir(emit_aperture_1, 1);
axis equal; view(3);
h_txAp = gcf;

figure ()

```

```

bpx_1 = sqrt(mean(pressure_tx_1.^2));
bpx_1 = bpx_1/max(bpx_1);
plot(mz*1000, bpx_1);
hold on;
bpx_2 = sqrt(mean(pressure_tx_2.^2));
bpx_2 = bpx_2/max(bpx_2);
plot(mz*1000, bpx_2);
hold on;
bpx_3 = sqrt(mean(pressure_tx_3.^2));
bpx_3 = bpx_3/max(bpx_3);
plot(mz*1000, bpx_3);
hold on;
title(sprintf('Transmit_beam_pressure_field'))
xlabel('Azimuth_position_[mm]');
ylabel('Normalized_axial_pressure_amplitude_');

```

7.0.9. Presión array rectangular con beam-forming por filas y diagrama de presión.

```

%% Comparación entre diferentes pitch(kerf's)
%% make sure that field_init has been called
eval('field_init(0)','1;')
clear;

%% DEFINE PARAMETERS
c = 750;
f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x_1 = 70e-6;
kerf_y_1 = 70e-6;
kerf_x_2 = 140e-6;
kerf_y_2 = 140e-6;
kerf_x_3 = 210e-6;
kerf_y_3 = 210e-6;
no_ele_x = 72;
no_ele_y = 9;
no_sub_x = 10;
no_sub_y = 10;
N_columns_enables = 15;
focus = [0 0 1.5]/1000;

%% FIND WHICH ELEMENTS TO USE

```

```

enabled=zeros(no_ele_x, no_ele_y);
enabled(29,:)=ones(1,no_ele_y);
enabled(30,:)=ones(1,no_ele_y);
enabled(31,:)=ones(1,no_ele_y);
enabled(32,:)=ones(1,no_ele_y);
enabled(33,:)=ones(1,no_ele_y);
enabled(34,:)=ones(1,no_ele_y);
enabled(35,:)=ones(1,no_ele_y);
enabled(36,:)=ones(1,no_ele_y);
enabled(37,:)=ones(1,no_ele_y);
enabled(38,:)=ones(1,no_ele_y);
enabled(39,:)=ones(1,no_ele_y);
enabled(40,:)=ones(1,no_ele_y);
enabled(41,:)=ones(1,no_ele_y);
enabled(42,:)=ones(1,no_ele_y);
enabled(43,:)=ones(1,no_ele_y);

%% DEFINE THE TRANSDUCER
emit_aperture_1 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_1,
    kerf_y_1, enabled, 1, 1, focus);
emit_aperture_2 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_2,
    kerf_y_2, enabled, 1, 1, focus);
emit_aperture_3 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_3,
    kerf_y_3, enabled, 1, 1, focus);

%% DELAYS
delay_times = zeros(1,N_columns_enables*no_ele_y);
xdc_focus_times(emit_aperture, 0, delay_times);
data = xdc_get(emit_aperture, 'focus');

%% Delaysrow_1
delaysrow_1 = [-1.6945e-07 -1.2971e-07 -9.4476e-08 -6.4196e-08 -3.9321e-08
    -2.0256e-08 -7.3403e-09 -8.1832e-10 -8.1832e-10 -7.3403e-09 -2.0256e-08
    -3.9321e-08 -6.4196e-08 -9.4476e-08 -1.2971e-07];

delay_times_1 = 1:no_ele_y*N_columns_enables;
m = 1;
while m <= no_ele_y*N_columns_enables
    for n = 1:N_columns_enables
        delay_times_1(m)= delaysrow_1(n);
        m = m+1;
    end
end

%% Delaysrow_2
delaysrow_2 = [-3.6945e-07 -2.8765e-07 -2.1306e-07 -1.4711e-07 -9.1428e-08

```

```

-4.7670e-08 -1.7425e-08 -1.9515e-09 -1.9515e-09 -1.7425e-08 -4.7670e-08
-9.1428e-08 -1.4711e-07 -2.1306e-07 -2.8765e-07];

delay_times_2 = 1:no_ele_y*N_columns_enables;
m = 1;
while m <= no_ele_y*N_columns_enables
    for n = 1:N_columns_enables
        delay_times_2(m)= delaysrow_2(n);
        m = m+1;
    end
end

%% Delaysrow_3
delaysrow_3 = [-6.2349e-07 -4.9329e-07 -3.7169e-07 -2.6126e-07 -1.6522e-07
-8.7518e-08 -3.2383e-08 -3.6511e-09 -3.6511e-09 -3.2383e-08 -8.7518e-08
-1.6522e-07 -2.6126e-07 -3.7169e-07 -4.9329e-07];

delay_times_3 = 1:no_ele_y*N_columns_enables;
m = 1;
while m <= no_ele_y*N_columns_enables
    for n = 1:N_columns_enables
        delay_times_3(m)= delaysrow_3(n);
        m = m+1;
    end
end

%% Procedure for creating a focus time line for an aperture

xdc_focus_times(emit_aperture_1, 0, delay_times_1);
data = xdc_get(emit_aperture_1, 'focus');

xdc_focus_times(emit_aperture_2, 0, delay_times_2);
data = xdc_get(emit_aperture_2, 'focus');

xdc_focus_times(emit_aperture_3, 0, delay_times_3);
data = xdc_get(emit_aperture_3, 'focus');

%% Definimos la excitaci n de la transmisi n de apertura.

Amplso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture_1, impulse_response);
xdc_impulse (emit_aperture_2, impulse_response);
xdc_impulse (emit_aperture_3, impulse_response);

```



```

%Excitaci n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture_1, excitation);
xdc_excitation (emit_aperture_2, excitation);
xdc_excitation (emit_aperture_3, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepth = 1.5/1000;
x_Start = -5/1000;
x_End = 5/1000;
Nmp = 1601;
mx = linspace(x_Start, x_End, Nmp)';
my = zeros(Nmp, 1);
mz = measDepth*ones(Nmp, 1);
measurement_points = [mx,my,mz];

%Para el colormap:
%Start = 0/1000;
%End = 4/1000;
%xStart = -4/1000;
%End = 4/1000;
%Nmpx = 150;
%Nmpz = 150;
%nx = linspace(xStart, xEnd, Nmpx)';
%ny = zeros(Nmpx*Nmpz, 1);
%nz = linspace(zStart, zEnd, Nmpz)';
%[X,Z] = meshgrid(mx,mz);
%measurement_points = [X(:),my,Z(:)];

%% Do the Calculation with Beamforming
%Transmit Pressure_1
[pressure_tx_1, startTime_tx_1] = calc_hp(emit_aperture_1, measurement_points);
%Transmit Pressure_2
[pressure_tx_2, startTime_tx_2] = calc_hp(emit_aperture_2, measurement_points);
%Transmit Pressure_3
[pressure_tx_3, startTime_tx_3] = calc_hp(emit_aperture_3, measurement_points);

%% PLOTS
figure ()
bpx_1 = sqrt(mean(pressure_tx_1.^2));
bpx_1 = bpx_1/max(bpx_1);
plot(mx*1000, bpx_1);
hold on;
bpx_2 = sqrt(mean(pressure_tx_2.^2));

```

```

bpx_2 = bpx_2/max(bpx_2);
plot(mx*1000, bpx_2);
hold on;
bpx_3 = sqrt(mean(pressure_tx_3.^2));
bpx_3 = bpx_3/max(bpx_3);
plot(mx*1000, bpx_3);
hold on;
title(sprintf('Beamprofile_at_depth=%0.3gmm', measDepth*1000))
xlabel('x_position_[mm]');
ylabel('Presi n_normalizada_');

%%Para el colormap:
%figure()
%bp = sqrt(mean(pressure_tx.^2));
%bp = reshape(bp, Nmpz, Nmpx);
%bp = bp/max(bp(:));
%pcolor(mx*1000,mz*1000, 20*log10(bp));
%shading interp
%title(sprintf('Transmit beam pressure field'))
%xlabel('Azimuth [mm]');
%ylabel('Range [mm]');
% %axis([-25 0]); %Set dynamic range
%colormap(jet(256));
%colorbar

```

7.0.10. Presión en en función de la profundidad en array rectangular con beam-forming individual.

```

%DEFINE PARAMETERS
c = 750;
f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x = 70e-6;
kerf_y = 70e-6;
no_ele_x = 72;
no_ele_y = 9;
no_sub_x = 10;
no_sub_y = 10;
N_columns_enables = 15;
focus = [0 0 1.5]/1000;

%%FIND WHICH ELEMENTS TO USE
enabled=zeros(no_ele_x, no_ele_y);

```

```

enabled(29,:)=ones(1,no_ele_y);
enabled(30,:)=ones(1,no_ele_y);
enabled(31,:)=ones(1,no_ele_y);
enabled(32,:)=ones(1,no_ele_y);
enabled(33,:)=ones(1,no_ele_y);
enabled(34,:)=ones(1,no_ele_y);
enabled(35,:)=ones(1,no_ele_y);
enabled(36,:)=ones(1,no_ele_y);
enabled(37,:)=ones(1,no_ele_y);
enabled(38,:)=ones(1,no_ele_y);
enabled(39,:)=ones(1,no_ele_y);
enabled(40,:)=ones(1,no_ele_y);
enabled(41,:)=ones(1,no_ele_y);
enabled(42,:)=ones(1,no_ele_y);
enabled(43,:)=ones(1,no_ele_y);

%% DEFINE THE TRANSDUCER
emit_aperture = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x, kerf_y,
    enabled, 1, 1, focus);

%% Definimos la excitaci3n de la transmisi3n de apertura.

%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture, impulse_response);

%Excitaci3n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepthStart = 0/1000;
measDepthEnd = 8/1000;
x = 0/1000;
Nmp = 1000;
mx = x*ones(Nmp,1);
my = zeros(Nmp, 1);
mz = linspace(measDepthStart,measDepthEnd,Nmp)';
measurement_points = [mx,my,mz];

%% Do the Calculation with Beamforming
%Transmit Pressure_1

```

```

[pressure_tx, startTime_tx] = calc_hp(emit_aperture, measurement_points);

%% PLOTS
% Ploteamos las columnas excitadas
figure()
grid on;
show_xdc_geir(emit_aperture, 1);
axis equal; view(3);
h_txAp =(gcf);

figure ()
bpx = sqrt(mean(pressure_tx.^2));
bpx = bpx/max(bpx);
plot(mz*1000, bpx);
title(sprintf('Beamprofile_along_depth_'));
xlabel('z_position_[mm]');
ylabel('Normal_axial_pressure_amplitude_');

```

7.0.11. Presión array rectangular con beam-forming individual y diagrama de presión.

```

%% Comparaci n entre diferentes pitch(kerf's)
%% make sure that field_init has been called
eval('field_init(0)','1;')
clear;

%% DEFINE PARAMETERS
c = 750;
f = 8e6;
fs = 1e9;
lambda = c/f;
height = 50e-6;
width = 50e-6;
kerf_x_1 = 70e-6;
kerf_y_1 = 70e-6;
kerf_x_2 = 140e-6;
kerf_y_2 = 140e-6;
kerf_x_3 = 210e-6;
kerf_y_3 = 210e-6;
no_ele_x = 72;
no_ele_y = 9;
no_sub_x = 10;
no_sub_y = 10;
N_columns_enables = 15;
focus = [0 0 1]/1000;

```

```

%% FIND WHICH ELEMENTS TO USE
enabled=zeros(no_ele_x, no_ele_y);
enabled(29,:)=ones(1,no_ele_y);
enabled(30,:)=ones(1,no_ele_y);
enabled(31,:)=ones(1,no_ele_y);
enabled(32,:)=ones(1,no_ele_y);
enabled(33,:)=ones(1,no_ele_y);
enabled(34,:)=ones(1,no_ele_y);
enabled(35,:)=ones(1,no_ele_y);
enabled(36,:)=ones(1,no_ele_y);
enabled(37,:)=ones(1,no_ele_y);
enabled(38,:)=ones(1,no_ele_y);
enabled(39,:)=ones(1,no_ele_y);
enabled(40,:)=ones(1,no_ele_y);
enabled(41,:)=ones(1,no_ele_y);
enabled(42,:)=ones(1,no_ele_y);
enabled(43,:)=ones(1,no_ele_y);

%% DEFINE THE TRANSDUCER
emit_aperture_1 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_1,
    kerf_y_1, enabled, 1, 1, focus);
emit_aperture_2 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_2,
    kerf_y_2, enabled, 1, 1, focus);
emit_aperture_3 = xdc_2d_array (no_ele_x, no_ele_y, width, height, kerf_x_3,
    kerf_y_3, enabled, 1, 1, focus);

%% Definimos la excitaci n de la transmisi n de apertura.

%Impulso
t_ir = -2/f:1/fs:2/f;
Bw = 0.4;
impulse_response = gauspuls(t_ir,f,Bw);
set_sampling(fs);
xdc_impulse (emit_aperture_1, impulse_response);
xdc_impulse (emit_aperture_2, impulse_response);
xdc_impulse (emit_aperture_3, impulse_response);

%Excitaci n
ex_periods = 4;
t_ex=(0:1/fs:ex_periods/f);
excitation=square(2*pi*f*t_ex);
xdc_excitation (emit_aperture_1, excitation);
xdc_excitation (emit_aperture_2, excitation);
xdc_excitation (emit_aperture_3, excitation);

%% DEFINE MEASUREMENT POINTS a lo largo de una linea
measDepth = 1/1000;

```

```

x_Start = -5/1000;
x_End = 5/1000;
Nmp = 1601;
mx = linspace(x_Start, x_End, Nmp)';
my = zeros(Nmp, 1);
mz = measDepth*ones(Nmp, 1);
measurement_points = [mx,my,mz];
%% Para el colormap
% zStart = 0/1000;
% zEnd = 4/1000;
% xStart = -4/1000;
% xEnd = 4/1000;
% Nmpx = 150;
% Nmpz = 150;
% mx = linspace(xStart, xEnd, Nmpx)';
% my = zeros(Nmpx*Nmpz, 1);
% mz = linspace(zStart, zEnd, Nmpz)';
% [X,Z] = meshgrid(mx,mz);
% measurement_points = [X(:),my,Z(:)];

%% Do the Calculation with Beamforming
%Transmit Pressure_1
[pressure_tx_1, startTime_tx_1] = calc_hp(emit_aperture_1, measurement_points);
%Transmit Pressure_2
[pressure_tx_2, startTime_tx_2] = calc_hp(emit_aperture_2, measurement_points);
%Transmit Pressure_3
[pressure_tx_3, startTime_tx_3] = calc_hp(emit_aperture_3, measurement_points);

%% PLOTS
figure ()
bpx_1 = sqrt(mean(pressure_tx_1.^2));
bpx_1 = bpx_1/max(bpx_1);
plot(mx*1000, bpx_1);
hold on;
bpx_2 = sqrt(mean(pressure_tx_2.^2));
bpx_2 = bpx_2/max(bpx_2);
plot(mx*1000, bpx_2);
hold on;
bpx_3 = sqrt(mean(pressure_tx_3.^2));
bpx_3 = bpx_3/max(bpx_3);
plot(mx*1000, bpx_3);
hold on;
title(sprintf('Beamprofile_at_depth_=%0.3gmm', measDepth*1000))
xlabel('x_position [mm]');
ylabel('Presi n_normalizada');

%% Para el colormap

```

```

% figure()
% bp = sqrt(mean(pressure_tx.^2));
% bp = reshape(bp, Nmpz, Nmpx);
% bp = bp/max(bp(:));
% pcolor(mx*1000,mz*1000, 20*log10(bp));
% shading interp
% title(sprintf('Transmit beam pressure field '))
% xlabel('Azimuth [mm]');
% ylabel('Range [mm]');
% %axis([-25 0]); % Set dynamic range
% colormap(jet(256));
% colorbar

```